

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

**Estudio y comparativa de herramientas generadoras de código para
aplicaciones Android**



AUTOR: Raúl García Pérez
DIRECTORA: María Francisca Rosique Contreras

Septiembre/ 2014



Autor	Raúl García Pérez
E-mail del Autor	garcia.perez.raul@gmail.com
Directora	María Francisca Rosique Contreras
E-mail del Director	paquirosique.upct@gmail.com
Título del PFC	Estudio y comparativa de herramientas generadoras de código para aplicaciones Android.
<p>Resumen</p> <p>El objetivo del presente proyecto final de carrera ha sido realizar un estudio y posterior comparativa de varias herramientas generadoras de código para aplicaciones Android. Estas herramientas permiten al usuario diseñar, desarrollar y gestionar aplicaciones Android sin tener conocimientos previos de programación.</p> <p>En primer lugar se ha realizado un estudio a nivel general de varias de estas herramientas, para posteriormente seleccionar dos de ellas con las que realizar un estudio más exhaustivo. Para ello se ha procedido a realizar una misma aplicación, a modo de ejemplo, llevada a cabo con sendas herramientas para así analizar y comparar diferentes aspectos.</p> <p>Es necesario tener en cuenta que estos servicios que desarrollan la aplicación por nosotros, tienen sus limitaciones en cuanto a complejidad, funcionalidad, alcance y diseño se refiere.</p>	
Titulación	Ingeniería Técnica de Telecomunicación, Especialidad Telemática.
Departamento	Tecnologías de la Información y las Comunicaciones (TIC)
Fecha de Presentación	Septiembre- 2014

Índice

CAPÍTULO 1.....	10
INTRODUCCIÓN	10
1.1 <i>Objetivos</i>	10
1.2 <i>Estructura del contenido</i>	10
CAPÍTULO 2.....	12
ANÁLISIS DE HERRAMIENTAS DE DESARROLLO	12
2.1 <i>Introducción a Android</i>	12
➤ Características y Especificaciones.....	14
➤ Arquitectura	16
2.2 <i>Estado del arte</i>	17
2.2.1 Introducción al desarrollo de aplicaciones Android	17
2.2.2 Herramientas disponibles para el desarrollo de aplicaciones	18
2.2.2.1 Desarrollo Clásico	19
2.2.2.2 Desarrollo Vía Web.....	20
2.3 <i>Comparativa de herramientas de desarrollo vía Web</i>	22
2.3.1 Comparativa de Funcionalidades entre Herramientas.....	26
2.3.2 Comparativa de Características entre Herramientas	27
CAPÍTULO 3.....	29
ANÁLISIS DE LAS HERRAMIENTAS SELECCIONADAS.	29
3.1 <i>Selección de herramientas para desarrollo de aplicación Android</i>	29
3.1.1 MIT App Inventor 2	31
3.1.1.1 Introducción a MIT App Inventor 2	31
3.1.1.2 Análisis de MIT App Inventor 2	32
➤ Instalación en Windows	32
➤ Entorno de diseño	32
➤ Editor de bloques	35
➤ Generación de la App	36
➤ Publicación de la App	38
3.1.2 Mobincube	38
3.1.2.1 Introducción a Mobincube	38
3.1.2.2 Análisis de Mobincube	39
➤ Instalación en Windows	39
➤ Entorno de diseño	40
➤ Generación de la App	47
➤ Publicación de la App	48
3.2 <i>Comparativa entre MIT App Inventor 2 y Mobincube</i>	49
CAPÍTULO 4.....	53
CASO PRÁCTICO - DESARROLLO DE APLICACIÓN AaSOS.	53
4.1 <i>Funcionalidad y objetivos de la aplicación AaSOS</i>	53
4.2 <i>Aplicación AaSOS</i>	54
4.2.1 Pantalla principal.....	54
4.2.2 Opciones de menú	54
➤ Emergencias 112	54
➤ Guardia Civil	55
➤ Mi Ubicación.....	55
➤ Alerta Peligro.....	56
➤ Batería Baja	57
➤ Manuales Emergencias.....	57
4.2.3 Diferencias de AaSOS con MIT App Inventor y Mobincube	58
4.2.3.1 Aplicación AaSOS con MIT App Inventor 2	58
4.2.3.2 Aplicación AaSOS con Mobincube	60
4.3 <i>Tutorial de desarrollo de AaSOS con MIT App Inventor 2</i>	64
4.4 <i>Tutorial de desarrollo de AaSOS con Mobincube</i>	84

CONCLUSIONES.....	98
ANEXOS	99
REFERENCIAS	109

Índice de Ilustraciones

<i>Ilustración 2.1: Android a primera vista</i>	<i>12</i>
<i>Ilustración 2.2: Arquitectura Android</i>	<i>17</i>
<i>Ilustración 2.3: Apps por sectores en Europa</i>	<i>21</i>
<i>Ilustración 2.4: Incremento de uso por tipos de aplicaciones móviles en España en 2013.....</i>	<i>21</i>
<i>Ilustración 2.5: Emulador de dispositivo móvil</i>	<i>25</i>
<i>Ilustración 3.1: Diagrama de funcionamiento de App Inventor</i>	<i>32</i>
<i>Ilustración 3.2: Entorno de diseño de App Inventor 2.....</i>	<i>33</i>
<i>Ilustración 3.3: Inventario de proyectos con MIT App Inventor 2.....</i>	<i>33</i>
<i>Ilustración 3.4: Selección de edición de bloques con MIT App Inventor 2</i>	<i>35</i>
<i>Ilustración 3.5: Editor de bloques con MIT App Inventor 2.....</i>	<i>35</i>
<i>Ilustración 3.6: Compilación con MIT App Inventor 2.....</i>	<i>37</i>
<i>Ilustración 3.7: Progreso de compilación con MIT App Inventor 2</i>	<i>37</i>
<i>Ilustración 3.8: Descarga de .apk con lectura de código QR en MIT App Inventor 2</i>	<i>38</i>
<i>Ilustración 3.9: Diagrama general Mobincube</i>	<i>39</i>
<i>Ilustración 3.10: Página de registro en Mobincube.....</i>	<i>39</i>
<i>Ilustración 3.11: Creación nueva app con Mobincube.....</i>	<i>40</i>
<i>Ilustración 3.12: Insertar nombre app con Mobincube</i>	<i>40</i>
<i>Ilustración 3.13: Selección de plantilla con Mobincube.....</i>	<i>41</i>
<i>Ilustración 3.14: Edición de información de la app con Mobincube</i>	<i>42</i>
<i>Ilustración 3.15: Selección de pantalla con Mobincube.....</i>	<i>42</i>
<i>Ilustración 3.16: Listado de pantallas con Mobincube</i>	<i>43</i>
<i>Ilustración 3.17: Esquema de pantallas en modo árbol con Mobincube.....</i>	<i>43</i>
<i>Ilustración 3.18: Acciones permitidas sobre una pantalla con Mobincube</i>	<i>44</i>
<i>Ilustración 3.19: Barra de herramientas de Mobincube</i>	<i>44</i>
<i>Ilustración 3.20: Propiedades generales de la aplicación con Mobincube</i>	<i>45</i>
<i>Ilustración 3.21: Propiedades comunes para todos los tipos de pantallas con Mobincube</i>	<i>46</i>
<i>Ilustración 3.22: Propiedades de la vista con Mobincube</i>	<i>46</i>
<i>Ilustración 3.23: Simulador con Mobincube</i>	<i>47</i>
<i>Ilustración 3.24: Compilación con Mobincube.....</i>	<i>47</i>
<i>Ilustración 3.25: Progreso de compilación con Mobincube.....</i>	<i>48</i>
<i>Ilustración 3.26: Descarga de .apk con lectura de código QR en Mobincube.....</i>	<i>48</i>
<i>Ilustración 3.27: Publicación de la aplicación con Mobincube</i>	<i>49</i>
<i>Ilustración 4.1: Logo de la aplicación AaSOS.....</i>	<i>53</i>
<i>Ilustración 4.2: Pantalla principal AaSOS</i>	<i>54</i>
<i>Ilustración 4.3: Llamada 112 con AaSOS</i>	<i>54</i>
<i>Ilustración 4.4: Llamada Guardia Civil con AaSOS.....</i>	<i>55</i>
<i>Ilustración 4.5.1: Envío de ubicación – Paso 1</i>	<i>56</i>
<i>Ilustración 4.5.2: Envío de ubicación – Paso 2</i>	<i>56</i>
<i>Ilustración 4.5.3: Envío de ubicación – Paso 3</i>	<i>56</i>
<i>Ilustración 4.6.1: Enviar aviso peligro – Paso 1</i>	<i>56</i>
<i>Ilustración 4.6.2: Enviar aviso peligro – Paso 2</i>	<i>56</i>
<i>Ilustración 4.7.1: Enviar aviso batería baja – Paso 1.....</i>	<i>57</i>
<i>Ilustración 4.7.2: Enviar aviso batería baja – Paso 2</i>	<i>57</i>
<i>Ilustración 4.8.1: Visualización de manuales – Paso 1.....</i>	<i>58</i>
<i>Ilustración 4.8.2: Visualización de manuales – Paso 2.....</i>	<i>58</i>
<i>Ilustración 4.9.1: Descarga de manuales en proceso</i>	<i>59</i>
<i>Ilustración 4.9.2: Descarga de manuales finalizada</i>	<i>59</i>
<i>Ilustración 4.10.1: Visualización de manuales con MIT App Inventor 2 – Paso 1</i>	<i>60</i>
<i>Ilustración 4.10.2: Visualización de manuales con MIT App Inventor 2 – Paso 2</i>	<i>60</i>
<i>Ilustración 4.11: Workaround de envío de ubicación</i>	<i>61</i>

<i>Ilustración 4.12.1: Envío de ubicación con workaround - Paso 1.....</i>	<i>61</i>
<i>Ilustración 4.12.2: Envío de ubicación con workaround - Paso 2.....</i>	<i>61</i>
<i>Ilustración 4.12.3: Envío de ubicación con workaround - Paso 3.....</i>	<i>61</i>
<i>Ilustración 4.13: Pantalla de inicio con publicidad de Mobincube.....</i>	<i>62</i>
<i>Ilustración 4.14: Banner de publicidad con Mobincube.....</i>	<i>62</i>
<i>Ilustración 4.15.1: Visualización de manuales con Mobincube – Paso 1.....</i>	<i>63</i>
<i>Ilustración 4.15.2: Visualización de manuales con Mobincube – Paso 2.....</i>	<i>63</i>
<i>Ilustración 4.15.3: Visualización de manuales con Mobincube – Paso 3.....</i>	<i>63</i>
<i>Ilustración 4.16: Tutorial MIT App Inventor 2 – Crear nuevo proyecto</i>	<i>64</i>
<i>Ilustración 4.17: Tutorial MIT App Inventor 2 – Asignar nombre nuevo proyecto.....</i>	<i>64</i>
<i>Ilustración 4.18: Tutorial MIT App Inventor 2 – Pantalla principal.....</i>	<i>64</i>
<i>Ilustración 4.19: Tutorial MIT App Inventor 2 – Editar icono, paso 1</i>	<i>65</i>
<i>Ilustración 4.20: Tutorial MIT App Inventor 2 – Editar icono, paso 2</i>	<i>65</i>
<i>Ilustración 4.21: Tutorial MIT App Inventor 2 – Editar icono, paso 3</i>	<i>65</i>
<i>Ilustración 4.22: Tutorial MIT App Inventor 2 – Crear pantalla Menú, paso 1</i>	<i>66</i>
<i>Ilustración 4.23: Tutorial MIT App Inventor 2 – Crear pantalla Menú, paso 2</i>	<i>66</i>
<i>Ilustración 4.24: Tutorial MIT App Inventor 2 – Crear pantalla Menú, paso 3</i>	<i>66</i>
<i>Ilustración 4.25: Tutorial MIT App Inventor 2 – Selección de imagen de fondo, paso 1.....</i>	<i>67</i>
<i>Ilustración 4.26: Tutorial MIT App Inventor 2 – Selección de imagen de fondo, paso 2.....</i>	<i>67</i>
<i>Ilustración 4.27: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 1.....</i>	<i>67</i>
<i>Ilustración 4.28: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 2.....</i>	<i>68</i>
<i>Ilustración 4.29: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 3.....</i>	<i>68</i>
<i>Ilustración 4.30: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 4.....</i>	<i>68</i>
<i>Ilustración 4.31: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 5.....</i>	<i>69</i>
<i>Ilustración 4.32: Tutorial MIT App Inventor 2 – Añadir Layout horizontal, paso 1.....</i>	<i>69</i>
<i>Ilustración 4.33: Tutorial MIT App Inventor 2 – Añadir Layout horizontal, paso 2.....</i>	<i>69</i>
<i>Ilustración 4.34: Tutorial MIT App Inventor 2 – Añadir Layout horizontal, paso 3.....</i>	<i>70</i>
<i>Ilustración 4.35: Tutorial MIT App Inventor 2 – Añadir opción de menú “Emergencias 112”, paso 1.....</i>	<i>70</i>
<i>Ilustración 4.36: Tutorial MIT App Inventor 2 – Añadir opción de menú “Emergencias 112”, paso 2.....</i>	<i>70</i>
<i>Ilustración 4.37: Tutorial MIT App Inventor 2 – Ajustar propiedades de botón “Emergencias 112”</i>	<i>71</i>
<i>Ilustración 4.38: Tutorial MIT App Inventor 2 – Añadir Layout y botón “Guardia Civil”</i>	<i>71</i>
<i>Ilustración 4.39: Tutorial MIT App Inventor 2 – Pantalla Menú completa</i>	<i>72</i>
<i>Ilustración 4.40: Tutorial MIT App Inventor 2 – Selección de objeto LocationSensor, paso 1</i>	<i>72</i>
<i>Ilustración 4.41: Tutorial MIT App Inventor 2 – Selección de objeto LocationSensor, paso 2</i>	<i>72</i>
<i>Ilustración 4.42: Tutorial MIT App Inventor 2 – Selección de objeto LocationSensor, paso 3</i>	<i>73</i>
<i>Ilustración 4.43: Tutorial MIT App Inventor 2 – Selección de objeto PhoneCall, paso 1.....</i>	<i>73</i>
<i>Ilustración 4.44: Tutorial MIT App Inventor 2 – Selección de objeto PhoneCall, paso 2.....</i>	<i>73</i>
<i>Ilustración 4.45: Tutorial MIT App Inventor 2 – Selección de objeto PhoneCall, paso 3.....</i>	<i>74</i>
<i>Ilustración 4.46: Tutorial MIT App Inventor 2 – Renombrar botones.....</i>	<i>74</i>
<i>Ilustración 4.47: Tutorial MIT App Inventor 2 – Acceso a editor de bloques</i>	<i>74</i>
<i>Ilustración 4.48: Tutorial MIT App Inventor 2 – Editor de bloques.....</i>	<i>75</i>
<i>Ilustración 4.49: Tutorial MIT App Inventor 2 – Evento inicialización Menú, paso 1.....</i>	<i>75</i>
<i>Ilustración 4.50: Tutorial MIT App Inventor 2 – Evento inicialización Menú, paso 2.....</i>	<i>75</i>
<i>Ilustración 4.51: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 1</i>	<i>76</i>
<i>Ilustración 4.52: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 2</i>	<i>76</i>
<i>Ilustración 4.53: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 3</i>	<i>76</i>
<i>Ilustración 4.54: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 4</i>	<i>77</i>
<i>Ilustración 4.55: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 5</i>	<i>77</i>
<i>Ilustración 4.56: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 6</i>	<i>77</i>
<i>Ilustración 4.57: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 7</i>	<i>78</i>
<i>Ilustración 4.58: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 8</i>	<i>78</i>
<i>Ilustración 4.59: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 9</i>	<i>78</i>
<i>Ilustración 4.60: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 10</i>	<i>79</i>
<i>Ilustración 4.61: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 11</i>	<i>79</i>
<i>Ilustración 4.62: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 12</i>	<i>79</i>
<i>Ilustración 4.63: Tutorial MIT App Inventor 2 – Añadir comentario, paso 1</i>	<i>80</i>

Ilustración 4.64: Tutorial MIT App Inventor 2 – Añadir comentario, paso 2	80
Ilustración 4.65: Tutorial MIT App Inventor 2 – Añadir comentario, paso 3	80
Ilustración 4.66: Tutorial MIT App Inventor 2 – Evento Click de botón “Emergencias112Button”, paso 1	81
Ilustración 4.67: Tutorial MIT App Inventor 2 – Evento Click de botón “Emergencias112Button”, paso 2	81
Ilustración 4.68: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 1	81
Ilustración 4.69: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 2	82
Ilustración 4.70: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 3	82
Ilustración 4.71: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 4	82
Ilustración 4.72: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 5	83
Ilustración 4.73: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 6	83
Ilustración 4.74: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 7	83
Ilustración 4.75: Tutorial Mobincube – Crear nuevo proyecto	84
Ilustración 4.76: Tutorial Mobincube – Asignar nombre nuevo proyecto	84
Ilustración 4.77: Tutorial Mobincube – Editar icono, paso 1	85
Ilustración 4.78: Tutorial Mobincube – Editar icono, paso 2	85
Ilustración 4.79: Tutorial Mobincube – Editar icono, paso 3	85
Ilustración 4.80: Tutorial Mobincube – Modificar datos app, paso 1	86
Ilustración 4.81: Tutorial Mobincube – Modificar datos app, paso 2	86
Ilustración 4.82: Tutorial Mobincube – Edición app	86
Ilustración 4.83: Tutorial Mobincube – Selección de pantalla inicial	87
Ilustración 4.84: Tutorial Mobincube – Selección de imagen de fondo, paso 1	87
Ilustración 4.85: Tutorial Mobincube – Selección de imagen de fondo, paso 2	87
Ilustración 4.86: Tutorial Mobincube – Selección de imagen de fondo, paso 3	88
Ilustración 4.87: Tutorial Mobincube – Selección de imagen de fondo, paso 4	88
Ilustración 4.88: Tutorial Mobincube – Selección de imagen de fondo, paso 5	88
Ilustración 4.89: Tutorial Mobincube – Selección de imagen de fondo, paso 6	89
Ilustración 4.90: Tutorial Mobincube – Selección de imagen de fondo, paso 7	89
Ilustración 4.91: Tutorial Mobincube – Selección de imagen de fondo, paso 8	89
Ilustración 4.92: Tutorial Mobincube – Selección de imagen de cabecera, paso 1	90
Ilustración 4.93: Tutorial Mobincube – Selección de imagen de cabecera, paso 2	90
Ilustración 4.94: Tutorial Mobincube – Selección de imagen de cabecera, paso 3	90
Ilustración 4.95: Tutorial Mobincube – Selección de imagen de cabecera, paso 4	91
Ilustración 4.96: Tutorial Mobincube – Selección de imagen de cabecera, paso 5	91
Ilustración 4.97: Tutorial Mobincube – Selección de imagen de cabecera, paso 6	91
Ilustración 4.98: Tutorial Mobincube – Selección de imagen de cabecera, paso 7	92
Ilustración 4.99: Tutorial Mobincube – Añadir opción de menú	92
Ilustración 4.100: Tutorial Mobincube – Añadir opción de menú “Emergencias 112”	92
Ilustración 4.101: Tutorial Mobincube – Asignar acción al botón “Emergencias 112”, paso 1	93
Ilustración 4.102: Tutorial Mobincube – Asignar acción al botón “Emergencias 112”, paso 2	93
Ilustración 4.103: Tutorial Mobincube – Asignar acción al botón “Emergencias 112”, paso 3	93
Ilustración 4.104: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 1	94
Ilustración 4.105: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 2	94
Ilustración 4.106: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 3	94
Ilustración 4.107: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 4	95
Ilustración 4.108: Tutorial Mobincube – Añadir opción de menú “Guardia Civil”, paso 1	95
Ilustración 4.109: Tutorial Mobincube – Añadir opción de menú “Guardia Civil”, paso 2	95
Ilustración 4.110: Tutorial Mobincube – Añadir todas las opciones del menú, paso 1	96
Ilustración 4.111: Tutorial Mobincube – Añadir todas las opciones del menú, paso 2	96
Ilustración 4.112: Tutorial Mobincube – Añadir todas las opciones del menú, paso 3	96
Ilustración 4.113: Tutorial Mobincube – Añadir todas las opciones del menú, paso 4	97
Ilustración 4.114: Tutorial Mobincube – Añadir todas las opciones del menú, paso 5	97
Ilustración 4.115: Tutorial Mobincube – Añadir todas las opciones del menú, paso 6	97
Ilustración Anexo I - 1: MIT App Inventor 2 - Vista de diseño de pantalla inicial	99
Ilustración Anexo I - 2: MIT App Inventor 2 - Vista de diseño de pantalla de menú	99
Ilustración Anexo I - 3: MIT App Inventor 2 - Vista de diseño de pantalla listado manuales	100

<i>Ilustración Anexo I - 4: MIT App Inventor 2 - Vista de bloques de pantalla inicial.....</i>	<i>102</i>
<i>Ilustración Anexo I - 5: MIT App Inventor 2 - Vista de bloques de pantalla de menú</i>	<i>104</i>
<i>Ilustración Anexo I - 6: MIT App Inventor 2 - Vista de bloques de pantalla listado manuales.....</i>	<i>106</i>
<i>Ilustración Anexo II - 1: Mobincube – Diseño de pantalla de menú.....</i>	<i>107</i>
<i>Ilustración Anexo II - 2: Mobincube - Diseño de pantalla listado manuales.....</i>	<i>107</i>
<i>Ilustración Anexo II - 3: Mobincube - Diseño de acción abrir manual.....</i>	<i>108</i>
<i>Ilustración Anexo II - 4: Mobincube - Diseño de pantalla para visualización de manual.....</i>	<i>108</i>

Índice de Tablas

<i>Tabla 2.1: Características y especificaciones actuales</i>	<i>16</i>
<i>Tabla 2.2: Comparativa de funcionalidades entre distintas herramientas.....</i>	<i>26</i>
<i>Tabla 2.3: Comparativa de características entre distintas herramientas (sector y sistemas operativos) .</i>	<i>27</i>
<i>Tabla 2.4: Comparativa de características generales entre distintas herramientas</i>	<i>28</i>
<i>Tabla 3.1: Comparativa de funcionalidades entre herramientas seleccionadas.....</i>	<i>30</i>
<i>Tabla 3.2: Comparativa de características entre herramientas seleccionadas</i>	<i>30</i>
<i>Tabla 3.3: Comparativa entre MIT App Inventor y Mobincube</i>	<i>52</i>

Capítulo 1

Introducción

1.1 Objetivos

El objetivo del presente proyecto final de carrera es realizar un estudio y posterior comparativa de varias herramientas generadoras de código para aplicaciones Android. Estas herramientas permiten al usuario diseñar, desarrollar y gestionar aplicaciones Android sin tener conocimientos previos de programación.

En primer lugar se ha realizado un primer estudio a nivel general de varias de estas herramientas, para posteriormente seleccionar dos de ellas con las que realizar un estudio más exhaustivo. Para ello se ha procedido a realizar una misma aplicación a modo de ejemplo llevada a cabo con sendas herramientas, analizando y comparando diferentes aspectos.

La elección de este estudio viene provocada por el vertiginoso auge experimentado en estos últimos años del uso de aplicaciones para dispositivos móviles.

Es parte clave para este estudio la de elegir tanto qué herramientas generadoras de código utilizar como la de qué aplicación implementar. Dependiendo de estas elecciones se orientará el resultado final del proyecto.

Al término del presente estudio se podrán comparar y valorar todos los aspectos fundamentales que hacen de cada herramienta generadora una herramienta distinta.

1.2 Estructura del contenido

El trabajo realizado en el proyecto final de carrera que se presenta se divide en los siguientes capítulos:

- **Capítulo 2.** Análisis de herramientas de desarrollo.

Se realizará una introducción a Android y se detallará un apartado de “estado del arte”. Además se realizará una clasificación de los diferentes tipos de herramientas generadoras de aplicaciones Android existentes y una comparativa de funcionalidades y características entre las mismas.

- **Capítulo 3.** Análisis de las herramientas seleccionadas.

Se realizará un análisis de las herramientas seleccionadas, justificando el porqué de su elección y evaluando una serie de propiedades de dichas herramientas.

Se describirá el proceso de creación, el entorno de desarrollo empleado y el software empleado para la realización de las apps con cada una de las herramientas.

Se mostrará una comparativa entre las herramientas seleccionadas, atendiendo a diferentes aspectos.

- **Capítulo 4.** Caso Práctico – Desarrollo de Aplicación AaSOS.

Se describirá la funcionalidad de la aplicación desarrollada, así como los objetivos a cumplir por la misma y se realizará una guía de uso de la aplicación con cada una de las herramientas. También se indicarán los inconvenientes encontrados durante el desarrollo y las diferencias entre ambas aplicaciones al ser creadas con distintas herramientas.

Capítulo 2

Análisis de Herramientas de Desarrollo

2.1 Introducción a Android

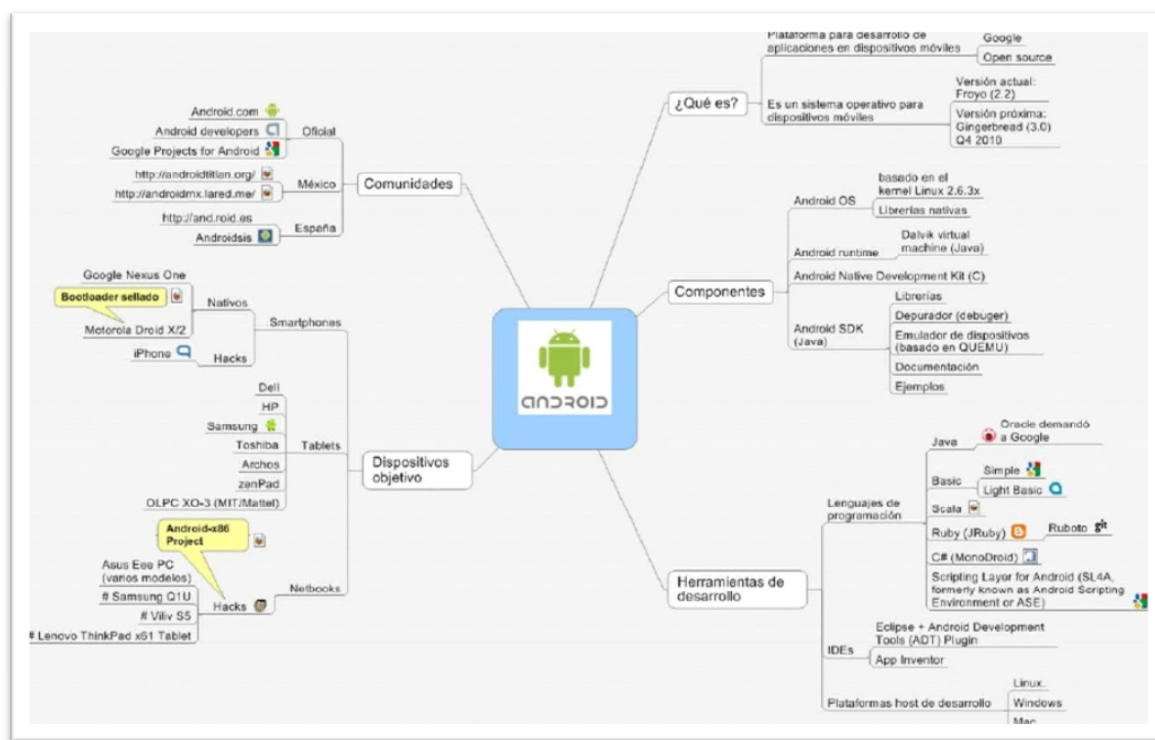


Ilustración 2.1: Android a primera vista

Android es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, TV, reproductores Mp3, PC'S, netbooks, automóviles, sistemas empujados y otros dispositivos. Su expansión ha sido espectacular, superando en la actualidad al S.O. Microsoft Windows.

Inicialmente desarrollado por Android, Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005. Android fue presentado en 2007 junto a la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

El éxito del sistema operativo se ha convertido en objeto de litigios sobre patentes en el marco de las llamadas «Guerras por patentes de teléfonos inteligentes» (en inglés Smartphone patent wars) entre las empresas de tecnología. Según documentos secretos filtrados en 2013 y 2014, el sistema operativo es uno de los objetivos de las agencias de inteligencia internacionales.

El 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, la compañía Google mostró un cambio de marca con el fin de unificar tanto el hardware como el software. Con ello mostraron nuevos productos como lo son Android TV, Android Auto, Android Wear o un Smartphone de baja gama llamado Android One. Con todo ello la marca se estabiliza para dar una imagen más firme tanto en el mercado como al público.

Android ha visto numerosas actualizaciones desde su liberación inicial. Estas actualizaciones al sistema operativo base típicamente arreglan bugs y agregan nuevas funciones.

La reiterada aparición de nuevas versiones que, en muchos casos, no llegan a funcionar correctamente en el hardware diseñado para versiones previas, hacen que Android sea considerado uno de los elementos promotores de la obsolescencia programada.

Android ha sido criticado muchas veces por la fragmentación que sufren sus terminales al no ser soportado con actualizaciones constantes por los distintos fabricantes.

Android es considerado como uno de los modelos de negocio más exitosos, pues su desarrollo estratégico contempla los factores que más se tienen en cuenta dentro de las herramientas y metodologías desarrollados por expertos en negocios. Este sistema operativo se ha convertido en un modelo a seguir por desarrolladores de tendencias y negocios de alto impacto.

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

En un principio, Android era eminentemente un sistema operativo pensado para usar con teclado, y gracias a un cursor poder navegar entre las aplicaciones. Desde su comienzo, Android ha sido altamente personalizable. Poco después, antes del lanzamiento del primer teléfono Android, esta filosofía cambió para convertirse en eminentemente táctil, y poder competir contra el iPhone, presentado 1 año y 9 meses antes.

En cuanto a la distribución de aplicaciones, Google Play es la tienda en línea de software desarrollado por Google para dispositivos Android y cuenta con unas 105.000 aplicaciones de las cuales el 62,2% son gratuitas.

Una aplicación llamada "Play Store" que se encuentra instalada en la mayoría de los dispositivos Android y permite a los usuarios navegar y descargar aplicaciones publicadas por los desarrolladores. Google retribuye a los desarrolladores el 70% del precio de las aplicaciones. Esta aplicación reemplazó a "Market".

Por otra parte, los usuarios pueden instalar aplicaciones desde otras tiendas virtuales (tales como Amazon Appstore o SlideME) o directamente en el dispositivo si se dispone del archivo APK de la aplicación.

➤ Características y Especificaciones

A continuación se detallan algunas de las características más importantes y especificaciones actuales:

<i>Diseño de dispositivo</i>	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
<i>Almacenamiento</i>	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
<i>Conectividad</i>	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX. GPRS, UMTS, HSPA+ Y HSDPA+
<i>Mensajería</i>	SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.
<i>Navegador web</i>	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3.
<i>Soporte de Java</i>	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.
<i>Soporte multimedia</i>	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.

<i>Soporte para streaming</i>	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
<i>Soporte para hardware adicional</i>	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
<i>Entorno de desarrollo</i>	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando el plugin de Herramientas de Desarrollo de Android.
<i>Google Play</i>	Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
<i>Multi-táctil</i>	Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa.
<i>Bluetooth</i>	El soporte para A2DP y AVRCP fue agregado en la versión 1.5 el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0 y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2. Los cambios incluyeron
<i>Video llamada</i>	Android soporta video llamada a través de Google Talk desde su versión HoneyComb.
<i>Multitarea</i>	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
<i>Características basadas en voz</i>	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
<i>Tethering</i>	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo

	PdaNet). Para permitir a un PC usar la conexión de datos del móvil android se podría requerir la instalación de software adicional.
--	---

Tabla 2.1: Características y especificaciones actuales

➤ Arquitectura

Los componentes principales del sistema operativo de Android son:

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Librerías:** Android incluye un conjunto de librerías de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El [núcleo](#) también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.



Ilustración 2.2: Arquitectura Android

2.2 Estado del arte

2.2.1 Introducción al desarrollo de aplicaciones Android

Con la llegada del “universo mobile” los usuarios de dispositivos móviles y aplicaciones crecen cada día superando ampliamente las expectativas.

La evolución de dichas aplicaciones nace con el lanzamiento del iphone de Apple y el desarrollo del sistema operativo para móviles Android. Junto a estos desarrollos llegan muchas más propuestas de smartphones, y de esta forma empieza el boom de las apps; juegos, noticias, diseño, arte, educación, fotografía, medicina... todo inmerso en lo que antes era un simple equipo de comunicación celular. La incorporación de internet en los celulares y la creación de las tablets revolucionó el mundo de las aplicaciones móviles. El surgimiento de las Apps Stores terminó de impulsar el éxito de las aplicaciones móviles y un significativo cambio en la manera en que se distribuye y comercializa el software.

El mercado de las aplicaciones móviles presenta un grandísimo potencial y los anunciantes comienzan a ser conscientes del partido que le pueden sacar.

La explicación de este éxito está en que con un Smartphone tienes Internet 24 horas al día los 7 días de la semana, lo que supone un acceso constante a la red; esto genera que el mercado de aplicaciones móviles no vaya a decrecer según lo observado. Mientras el número de celulares inteligentes y tablets siga creciendo, la demanda de aplicaciones seguirá aumentando.

A día de hoy, Android dispone de cientos de millones de dispositivos móviles en más de 190 países a lo largo del mundo. Y todo ello acompañado de más de 1.5 billones de descargas de aplicaciones desde Google Play cada mes. Unos números asombrosos que a muchas personas les han llevado a querer aportar su granito de arena, desarrollando su propia aplicación.

En España, 20 millones de personas descargan y usan aplicaciones cada día superando el tiempo que los usuarios pasan en webs. La media de tiempo que estos usuarios usan internet en su dispositivo móvil es de 75 minutos/día.

En Alemania, Francia, Italia, Inglaterra y España, el 12,3% de los usuarios con móvil también han declarado tener una tablet. España, lidera este grupo de países con un 17,3%.

Las aplicaciones móviles o “apps” han experimentado un incremento del 115% en su uso en el pasado 2013. En concreto, las aplicaciones relacionadas con mensajería instantánea y con redes sociales han sido las más populares entre los usuarios, alcanzando un aumento del 203% en el mismo año. Después de las mencionadas “apps” de mensajería y entornos sociales, las de servicios públicos y orientadas a productividad han sido las más usadas con un aumento del 149% respecto a 2013.

A estas aplicaciones les siguen en el ranking las “apps” de uso general, las de música y entretenimiento, las de compras y las de juegos, en ese orden. Finalmente, esta lista termina con las aplicaciones relacionadas con el deporte y la salud, y las de periódicos y revistas como las que menos han aumentado su uso en 2013, con un incremento del 49% y 31% respectivamente.

Todos estos incrementos son achacados en gran medida a la edad media de la inmensa mayoría de los usuarios de estas “apps”, que se encuentra entre los 16 y 31 años de edad.

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios marcos de aplicaciones basadas en la web multiteléfono. También es posible usar las bibliotecas Qt gracias al proyecto Necesitas SDK.

2.2.2 Herramientas disponibles para el desarrollo de aplicaciones

Hoy en día el teléfono móvil se ha convertido en una herramienta imprescindible, que ya nos acompaña siempre allí donde vayamos.

Es un hecho que los consumidores están más familiarizados con la utilización de la tecnología móvil que incluso con otras tecnologías más maduras. Gradualmente los

terminales móviles están siendo equipados con una gama mucho más amplia de funciones que nos permiten responder a las demandas de la movilidad, convirtiéndose en un instrumento similar a un PC.

Por ello, cada vez son más las aplicaciones móviles y soluciones wireless - basadas en los nuevos usos del teléfono móvil u otros dispositivos móviles avanzados como PDAs y Palmtops-, reemplazan de manera eficiente a las herramientas, procesos y modelos anteriores y se extienden rápidamente en el entorno empresarial actual.

Al instalar una aplicación es posible que nos pidan autorización para que permitamos que se acceda a la información de nuestro dispositivo. Desde algunas aplicaciones se puede acceder a:

- Lista de contactos de teléfono y de email.
- Al registro de llamadas.
- A los datos transmitidos/recibidos por internet.
- A la información del calendario.
- Al contenido multimedia.
- A los datos de localización del aparato.
- Al código de identificación exclusivo del dispositivo.
- A información que indica la manera en que se usa la aplicación propiamente dicha.
- A otros proveedores o aplicaciones.
- Elementos tanto hardware como software del dispositivo.
- SMS
- Etc

Algunas aplicaciones solamente pueden acceder a los datos necesarios para su funcionamiento, otras pueden acceder a datos que no están relacionados con el propósito de la aplicación.

Dependiendo del tipo de aplicación que se quiere realizar, es muy importante valorar si la herramienta te permite conectar y gestionar los recursos del propio dispositivo móvil (cámara, gps, archivos, agenda, etc), ya que de esta manera permite una mayor interacción con el usuario y una mejor experiencia interactiva.

En cuanto a Sistemas Operativos se refiere, Android es el líder destacado con un casi 55% del total de dispositivos móviles en uso.

2.2.2.1 Desarrollo Clásico

El método clásico para el desarrollo de aplicaciones Android requiere la instalación y correspondiente configuración del IDE. Se requiere la instalación de:

- Java JDK 6
- Android SDK 2.2 o superior
- Eclipse 3.4 o superior
- Android Development Toolkit (ADT) – Plugin para eclipse.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable

de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente.

➤ Android SDK

El SDK (Software Development Kit) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales. Las plataformas de desarrollo soportadas incluyen Linux (cualquier distribución moderna), Mac OS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo (IDE, Integrated Development Environment) soportada oficialmente es Eclipse junto con el complemento ADT (Android Development Tools plugin), aunque también puede utilizarse un editor de texto para escribir ficheros Java y Xml y utilizar comandos en un terminal (se necesitan los paquetes JDK, Java Development Kit y Apache Ant) para crear y depurar aplicaciones. Además, pueden controlarse dispositivos Android que estén conectados (e.g. reiniciarlos, instalar aplicaciones en remoto).

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad.

Una aplicación Android está compuesta por un conjunto de ficheros empaquetados en formato .apk y guardada en el directorio /data/app del sistema operativo Android (este directorio necesita permisos de superusuario, root, por razones de seguridad). Un paquete APK incluye ficheros .dex (ejecutables Dalvik, un código intermedio compilado), recursos, etc. Cualquier aplicación comprimida en formato APK, puede instalarse sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

2.2.2.2 Desarrollo Vía Web

Crear aplicaciones para móviles actualmente es muy caro y requiere mucho tiempo, pero con este tipo de herramientas web esto ha cambiado. Son plataformas (generalmente multi-idioma) que permiten que todas las personas sean capaces de diseñar, desarrollar y gestionar sus propias aplicaciones de una manera fácil y rápida sin necesidad de tener conocimientos previos de programación y con un resultado personalizado y profesional.

Existen en el mercado multitud de servicios que permiten crear desde cero y paso a paso aplicaciones para las diferentes plataformas móviles, utilizando los asistentes de edición para personalizar el diseño de la interfaz y configurar las funciones de aplicaciones básicas. Una vez terminada nuestra creación, los mismos asistentes se encargan de compilar nuestro proyecto en una aplicación nativa para la plataforma elegida, guiándonos finalmente en el proceso de publicación de la misma en la tienda de aplicaciones.

En la mayoría de los casos el desarrollo y publicación de estas aplicaciones está enfocado al ámbito comercial, todo aquel que posee un negocio quiere difundirlo y publicitarlo al máximo. El uso de aplicaciones es una vía cada vez con más auge para este fin. Con estas herramientas se facilita la creación de las mismas, ya que no supone prácticamente coste alguno.

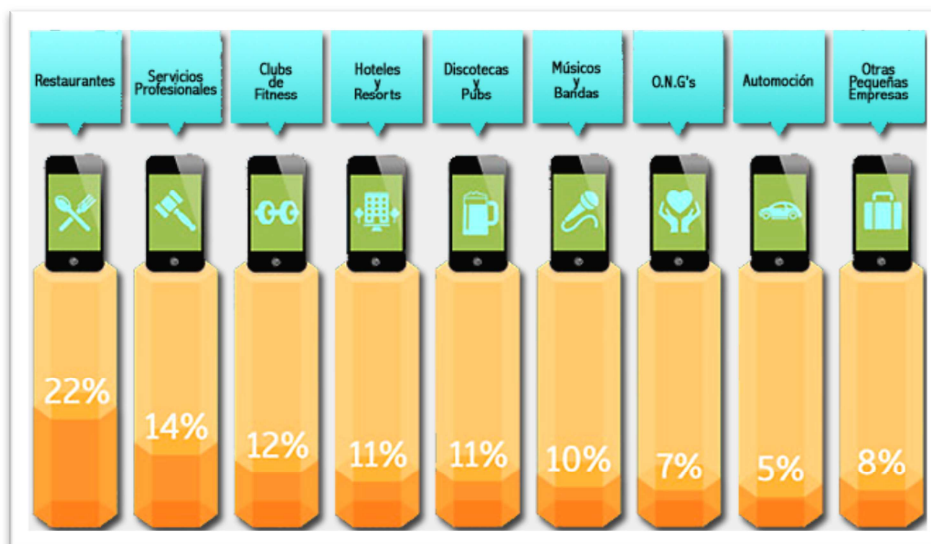


Ilustración 2.3: Apps por sectores en Europa

También existe un sector importante, en cuanto al tipo de uso de las aplicaciones móviles se refiere, enfocado a utilidades comunes, comunicación, entretenimiento, información y juegos.

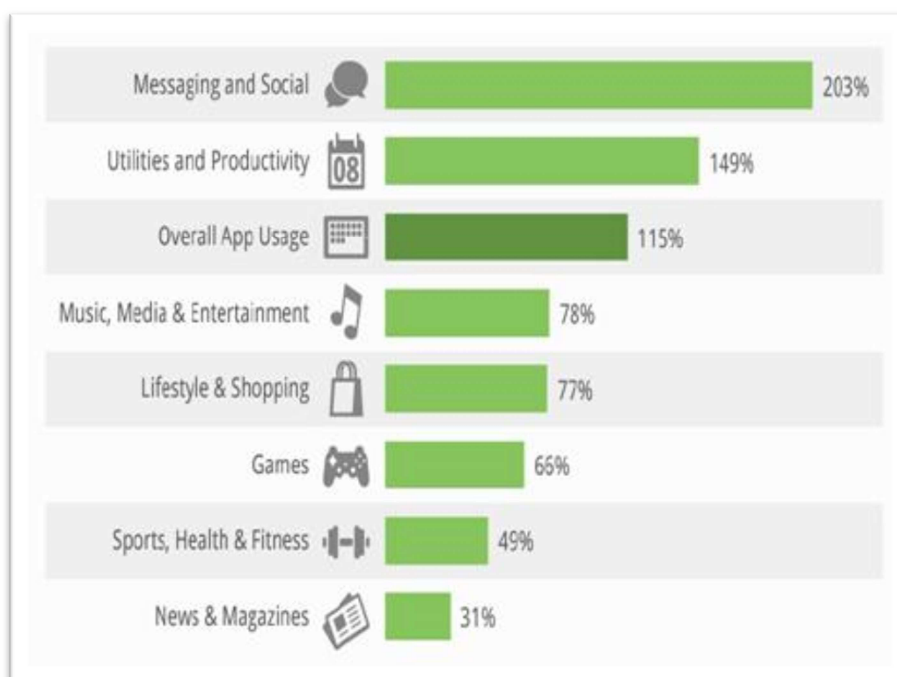


Ilustración 2.4: Incremento de uso por tipos de aplicaciones móviles en España en 2013

Aun teniendo en cuenta las numerosas ventajas de las aplicaciones móviles creadas vía web, también hay que hablar de las desventajas que estas acarrearán. Por lo general, todas ellas poseen importantes limitaciones en sus funciones, pues éstas han de estar desarrolladas previamente y suelen ser las más básicas o usuales. A pesar de ese inconveniente, lo cierto es que poder crear una aplicación móvil, sin poseer ningún tipo de conocimiento de diseño o programación, es una auténtica revolución. Crear nuestra propia app está al alcance de cualquiera.

2.3 Comparativa de herramientas de desarrollo vía Web

En el presente apartado, se va a realizar más a fondo una comparativa entre distintas herramientas de desarrollo de aplicaciones vía web.

Referente al tema de limitaciones en las funciones que estas herramientas proporcionan, hay diferentes escalafones en los que clasificarlas, ya que la complejidad o potencia de la aplicación viene derivada de la complejidad y potencial de la herramienta; y la posibilidad de manejo de esta viene limitada por los conocimientos de programación del usuario. Es decir, a mayor conocimiento del usuario en programación, mejor posibilidad de elegir una herramienta más completa y potente, y por lo tanto, mayor posibilidad de crear una aplicación robusta y con más posibilidades.

Algunos de los aspectos más destacados por los que se pueden clasificar estas herramientas se detallan a continuación, aunque en los siguientes apartados (2.3.1 y 2.3.2) se muestran a modo de tablas unas comparativas de funcionalidades y características ampliando dicha información:

➤ Gratuidad o no de las herramientas

La mayoría de las herramientas ofrecen la posibilidad de comprar diferentes paquetes (con diferentes precios) los cuales amplían las funcionalidades y potencia de la herramienta, pudiendo así crear aplicaciones más completas y complejas dependiendo del paquete adquirido. En este caso se ha trabajado en todo momento con versiones 100% gratuitas.

El precio de los paquetes se mueve en un margen que va desde ser totalmente gratuita ofreciendo todas las funcionalidades que posee la herramienta en cuestión, pasando por unos pocos euros al mes y pudiendo llegar hasta más de 159 €/mes.

Estos son algunas de las mejoras y posibilidades que se ofrecen con la compra de dichos paquetes:

- N° de mensajes Push
- Programas de fidelización
- Cupones QR
- Lector QR
- Cupones GPS
- Tienda móvil

- Gestor de pedidos
- Gestor de reservas
- Direcciones GPS
- Puntos de interés
- Integración con otros proveedores
- Mapas
- Botón de llamada
- Actualizaciones ilimitadas
- Páginas contenido HTML
- Contenido multimedia
- Calendario de eventos
- Podcast
- Youtube
- Potentes estadísticas de uso
- Código QR para distribución
- Publicación en Google Play
- Publicación en AppStore
- N° de apps
- Espacio de almacenamiento
- Eliminar publicidad
- Quitar intro
- Espacio en Cloud
- Colas de prioridad para los trabajos en construcción
- Gestión de ingresos por publicidad o ventas
- Modificar la aplicación cuantas veces se desee una vez publicada
- Ver los cambios y actualizarla en tiempo real

Existen algunas herramientas que te permiten trabajar con una versión de prueba con una duración limitada o también las que te permiten trabajar gratuitamente con la máxima funcionalidad y sólo se pagaría en el momento de publicar la aplicación. Otra modalidad es que te sea permitido crear gratuitamente las primeras aplicaciones que realices, pasado ese límite habría que adquirir algún paquete para poder seguir trabajando.

➤ Posibilidades de creación

Para empezar a crear una aplicación estas herramientas nos ofrecen 2 modos distintos de hacerlo:

- Desde cero: el usuario no posee nada hecho teniendo así que empezar desde el principio.
- A partir de cualquier contenido web ya existente: el usuario puede convertir cualquier página o aplicación web en una aplicación móvil. Para este proceso se tardaría menos tiempo que para el anterior ya que se dispone de todos los contenidos necesarios y de una estructura con la que empezar a trabajar. La aplicación puede ser “clonada” de la web o realizársele modificaciones.
Ejemplo: <http://www.appsgeyser.com>

➤ **Tipo de creación**

- Creación de código por módulos autocompilables: se crea la aplicación usando un entorno gráfico a base de bloques a modo de pseudocódigo. Esto es autocompilado posteriormente por la herramienta siendo este proceso transparente para el usuario.
Ejemplo: <http://appinventor.mit.edu/explore>
Se requiere: Ordenador con cualquier sistema operativo, conocimientos previos de programación a nivel medio.
- Usando plantillas ofrecidas por la herramienta: la mayoría de herramientas usan esta modalidad. Posee una muy buena relación entre la complejidad para crear una aplicación, ya que apenas se requieren conocimientos previos de programación. Si unimos esto a la gran cantidad de plantillas ofrecidas que abarcan prácticamente cualquier ámbito comercial y la calidad del resultado obtenido, descubrimos el motivo de por qué se trata del tipo de herramienta usada más extendida entre el total de usuarios de este tipo de herramientas.
Ejemplo: <http://apps.apps-builder.com/wizard2/step1/763307>
Se requiere: Ordenador con cualquier sistema operativo.
- Panel de control en línea: en este modo primero se crean las pantallas, después se descarga el código que genera la herramienta para posteriormente compilarlo en un entorno de desarrollo (como puede ser Eclipse).
Ejemplo: <https://www.buzztouch.com/pages/howitworks.php>
Se requiere: Ordenador con cualquier sistema operativo, un entorno de desarrollo, kit de desarrollo de aplicaciones android (SDK) para aplicaciones móviles y conocimientos previos de programación a nivel bajo.
- Creando la aplicación desde el propio dispositivo móvil: se trata de lo último que ha surgido. Existe ya la posibilidad de realizar la creación de aplicaciones desde el propio dispositivo móvil. Para ello sólo hay que descargar la aplicación de cualquier tienda de software en línea y seguir las instrucciones que se indican.
Ejemplo: <http://www.rheti.com>
Se requiere: un dispositivo móvil con conexión a Internet.

➤ Tipos de visualización/pruebas

Durante el proceso de creación existen dos opciones para poder visualizar la aplicación e ir realizando pruebas del funcionamiento existen dos opciones:

- Desde un dispositivo móvil Android: conectando dicho dispositivo al ordenador donde se está trabajando.
- Emulador: la herramienta te ofrece elegir un emulador de un dispositivo móvil.



Ilustración 2.5: Emulador de dispositivo móvil

➤ Posibilidad de publicar la app:

Prácticamente en todas las herramientas se ofrece al usuario la posibilidad de publicar su aplicación en tiendas de software en línea como lo son Google Play o App Store. Prácticamente en todas las herramientas utilizadas es aquí donde se exige el pago de alguna cuota.

➤ Ámbito de uso

Se trata de un factor muy importante a la hora de elegir la herramienta con la que crear la aplicación deseada.

Esta es una clasificación general de las mismas:

- Mensajería y redes sociales. Han experimentado el mayor crecimiento global en los últimos años, mostrándose casi indispensables para la mayoría de sus usuarios.
- Utilidades y productividad. Gran cantidad de variedad de aplicaciones de este tipo que ayudan y facilitan la vida al usuario de muy diferentes formas.

- Aplicación de uso general. Aquí se engloban aplicaciones como procesadores de texto, editores, hojas de cálculo, sistemas gestores de bases de datos, programas de diseño asistido, etc.
- Multimedia: musica, contenido media y entretenimiento.
- Comercios y tiendas. Destinadas a publicitar el establecimiento y/o dar a conocer e informar al usuario de las características del mismo.
- Juegos. Son una parte muy importante del total de descargas de apps, llegando en España hasta el 34% del total de descargas durante el año 2013.
- Deportes y salud. Gestores de calorías, cuenta kilómetros, dietarios. Entre las más conocidas y usadas están Niké + o Endomondo.
- Noticias, periódicos y revistas. Mantienen al usuario informado de todo lo que le pueda interesar.

2.3.1 Comparativa de Funcionalidades entre Herramientas

Se muestra una tabla comparativa de funcionalidades entre algunas de las herramientas más populares para el desarrollo de aplicaciones Android sin conocimientos de programación.

Herramienta / Funcionalidad	Adiante apps	Andromo	Appmaker	Apps Builder	AppsGeyser	Appyet	Buzztouch	Creapp	Ibuddapp	MIT App Inventor 2	Mobineube	Nimbo solutions	Octopus	Rhedi	Shoutem	Wiiappii	Y app
Compartir contenido	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗
Envío de SMS	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗
Realización de llamadas	✓	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗
Ubicación	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✗	✓	✗	✗	✗
Visualización PDF	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
Envío de correos	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗
Acceso a contenido de la SD	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
Calendario eventos	✓	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
Cloud	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗
Código QR distribución	✗	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
Integración con redes sociales	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Contenido HTML	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗
Contenido multimedia	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Descargas	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✓	✓
Encuesta	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓
Estadísticas de uso	✓	✗	✓	✗	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗	✓	✗
Gestor de pedidos	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
Gestor de reservas	✓	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
GPS	✗	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗
Lector QR	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Mapas	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗
Notificaciones Push	✓	✗	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Monetización	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗
Soporte	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Tienda móvil	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Uso cámara del dispositivo	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗
Uso de otros proveedores	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Video en directo	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Youtube	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓
Web embebida	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
RSS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
Podcasting	✗	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
Navegación web	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
Aplicaciones ilimitadas	✗	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓

Tabla 2.2: Comparativa de funcionalidades entre distintas herramientas

2.3.2 Comparativa de Características entre Herramientas

A continuación se muestran unas tablas comparativas con información relevante sobre características de las mismas aplicaciones a las que se les ha realizado el estudio de funcionalidades:

➤ Sistemas operativos y sector al que van dirigidas

Característica Herramienta	Sistemas operativos	Sector al que va dirigido
MIT app Inventor	Android	Solución integral para autónomos y PYMES
Mobiincube	iOS, Android, Windows phone	Destinada a PYMES, autónomos y particulares sin conocimientos de programación
Buzztouch	iOS, Android	El servicio va dirigido a todo tipo de clientes: desarrolladores, negocios o cualquier persona interesada en tecnología móvil
Adiante apps	iOS, Android	PYMES. Especialmente negocios en sectores como hostelería o inmobiliarias
Andromo	Android	Especialmente destinado a PYMES y autónomos
Appmakr	iOS, Android, HTML5	Especialmente para crear aplicaciones Android sencillas enfocadas al contenido web o de blog
Apps Builder	iOS, Android, HTML5	Aplicaciones para PYMES o juegos en 3D
AppsGeyser	iOS, Android	Especialmente destinada al desarrollo aplicaciones Android sencillas enfocadas a mostrar blogs o contenido web. Para los negocios en línea que se ocupan de sus usuario y la marca, permitiendo establecer un canal de comercialización para la comunicación y atracción de nuevos usuarios.
Appyet	Android	Convierte sitios webs en aplicaciones nativas para Android. Especialmente destinada a aplicaciones para blogs y podcast
Creapp	iOS, Android, HTML5	Destinada a proyectos de pedidos en línea, pagos por Internet y más funcionalidades perfectas para pequeñas y medianas empresas...
Game Salad	Android, iOS, Mac, HTML5 y Windows 8	Desarrollo de juegos
Ibuildapp	iOS, Android, HTML5	Especialmente destinada a PYME Empresarios o directivos.
Nimbo solutions	iOS, Android	Desarrolladores de aplicaciones móviles. Desarrolladores web.
Octopus	iOS, Android, Windows Phone	Aplicaciones móviles para comercios locales y PYMES
Rheti	Android	A día de hoy, Rheti se encuentra en fase beta y sólo es posible su descarga si nos damos de alta en la comunidad de Rheti para poder probar sus funcionalidades. Aún así desde su página web nos aseguran que en breve lanzarán una versión beta pública que estará disponible en Google Play.
Shoutem	iOS, Android, HTML5	Destinada a particulares, autónomos o PYMES. Permite crear elegantes aplicaciones con HTML5 que utilizan como fuente de información blogs, cuentas sociales, etc.

Tabla 2.3: Comparativa de características entre distintas herramientas (sector y sistemas operativos)

➤ Características generales

Característica Herramienta	Conocimientos previos de programación	Tiempo de documentación para desarrollo de app básica (minutos)	Gratuidad	Ampliación de funcionalidades con versión de pago	Nº de Apps gratuitas	Nº de Apps permitidas previo pago	Posibilidad de publicación (implica pago)	Monetización	Publicidad en versiones gratuitas	Requerimientos técnicos: instalación de software, apps, etc	Autocompilación	Modificación de App una vez publicada	Almacenamiento de datos en el teléfono	Plantillas predefinidas	Posibilidad de creación desde cero
MIT app Inventor	Minimos	30	✓	✗	Ilimitadas	N/A	✗	✗	✗	✓	✓	✗	✓	✗	✓
Mobincube	✗	10	✓	✓	Ilimitadas	N/A	✓	✓	✓	✗	✓	✓	✗	✓	✓
Buzztouch	✗	60	✓	✗	3	50	✗	✓	✗	✓	✗	✓	✗	✓	✗
Adiante apps	✗	20	✓	✓	Ilimitadas	1	Si, 1 app	✓	✗	✓	✓	✓	✗	✓	✓
Andromo	✗	15	✓	✓	1	50 o ilimitadas	✗	✓	✓	✗	✓	✓	✗	No. Bloques	✓
Appmakr	HTM, CSS	15	✓	✓	Ilimitadas	1	✓	✓	✓	✗	✓	✗	✗	✗	✓
Apps Builder	✗	15	No. Prueba 30 días	✓	Ilimitadas (pruebas 14 días)	1	✓	✓	N/A	✗	✓	✓	✗	✓	✗
AppsGeyser	✗	15	✓	N/A	Ilimitadas	N/A	✗	✓	✓	✗	✓	✗	✗	✓	✗
Appyet	✗	5	✓	N/A	Ilimitadas	N/A	✓	✓	✗	✗	✓	✓	✗	Si, creadas	✓
Creapp	✗	15	✓	✗	Ilimitadas	N/A	✓	✓	✗	✗	✓	✓	✗	✓	✓
Game Salad	✗	240	✓	✓	Ilimitadas	N/A	✓	✓	✗	✓	✓	✗	✗	✓	✓
Ibuildapp	✗	15	✓	✓	Ilimitadas	N/A	✓	✓	✗	✗	✓	✗	✗	✓	✓
Nimbo solutions	✗	15	✓	✓	Ilimitadas	N/A	✓	✓	✓	Si, en el dispositivo movil	✓	✓	✗	✓	✗
Octopus	✗	15	✓	✓	1	Ilimitadas	✓	✓	✓	✗	✓	✓	✗	✓	✓
Rheti	✗	15	✓	✓	Ilimitadas	N/A	✓	✗	✗	✓	✓	✗	✗	✓	✓
Shoutem	✗	15	No. Prueba Si (contenido web)	✓	Ninguna	Ilimitadas	✓	✓	N/A	✗	✓	✗	✗	✓	✓
Wiiappii	✗	15	✓	✓	1	1	✓	✓	✓	✗	✓	✓	✗	✓	✓

Tabla 2.4: Comparativa de características generales entre distintas herramientas

Capítulo 3

Análisis de las Herramientas Seleccionadas.

3.1 Selección de herramientas para desarrollo de aplicación Android

Una vez realizado el análisis de herramientas disponibles para la generación de aplicaciones Android, se han seleccionado 2 de ellas y se ha realizado la misma aplicación con ambas, con el objetivo de poder realizar una comparativa más exhaustiva entre ellas.

Tal y como se ha comentado en el capítulo anterior, la elección de estas herramientas es consecuencia directa del potencial de las mismas y de la funcionalidad que se quiere dar en la aplicación a desarrollar.

Algunos de los aspectos tenidos en cuenta han sido:

- Conocimientos previos
- Gratuidad de la herramienta
- Funcionalidades permitidas
- Autocompilación
- Posibilidad de creación desde cero.
- Etc.

Uno de los puntos clave ha sido pensar en qué tipo de aplicación desarrollar para el presente proyecto final de carrera. La inmensa mayoría de herramientas tienen una funcionalidad muy básica destinada a mostrar contenido estático precargado previamente en la aplicación (destinadas a ámbito comercial en gran parte: restaurantes, tiendas, hoteles, gimnasios, etc) mediante el uso de plantillas y no permiten realizar aplicaciones complejas (configurar mensajes, acceder a la agenda de contactos, trabajar con otras apps del dispositivo, etc) aunque si completas.

Pensando en la realización de una aplicación que permitiera explotar al máximo el potencial de estas herramientas, siempre en su versión gratuita y dejando al margen la visualización de contenido estático, las funcionalidades con las que se quiere dotar a la aplicación son las siguientes:

- Compartir Ubicación y mensajes de texto
 - SMS
 - WhatsApp
 - Email
 - ...
- Realizar llamadas telefónicas
- Visualización de Pdf's

En el capítulo 4 se explicará con más detalle la aplicación desarrollada, así como los problemas que surgieron durante el desarrollo, pero para conocer el porqué de la elección de estas 2 herramientas bastaría con conocer las funcionalidades requeridas.

Basándonos en las tablas de características y funcionalidades permitidas por las herramientas analizadas, vamos a llevar a cabo la selección de las herramientas con las que se realizará el desarrollo de la aplicación.

Si nos fijamos en la tabla de funcionalidades podríamos decir que 3 aplicaciones se adaptarían a las necesidades de funcionalidades requeridas por la aplicación a desarrollar. Todas ellas tienen las funcionalidades necesarias, excepto Mobincube que no permite la visualización directa de Pdf's, sin embargo, existen alternativas que permitirán realizar esta acción. Por otro lado MIT App Inventor 2 tiene una funcionalidad adicional que es la de poder acceder a contenido de la tarjeta SD del dispositivo. Esta funcionalidad de acceso a la memoria SD del dispositivo, nos será útil a la hora de visualizar Pdf's almacenados en el dispositivo.

Herramienta / Funcionalidad	Adiante apps	Andromo	Appmaker	Apps Builder	AppsGeyser	Appyret	Buzztouch	Creapp	Ibuildapp	MIT App Inventor 2	Mobincube	Nimbo solutions	Octopus	Rheti	Shoutem	Wtiappii	Y app
Compartir contenido	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗
Envío de SMS	✗	✗	✗	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗
Realización de llamadas	✓	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Ubicación	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗	✓	✗
Visualización PDF	✗	✓	✗	✓	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Envío de correos	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗
Acceso a contenido de la SD	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Calendario eventos	✓	✗	✗	✓	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓
Cloud	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓
Código QR distribución	✗	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗

Tabla 3.1: Comparativa de funcionalidades entre herramientas seleccionadas

Si continuamos analizando la tabla de características, son varios los factores que nos hacen descartar la herramienta Buzztouch:

Característica / Herramienta	Conocimiento previo de programación	Tiempo de documentación para desarrollo de app basica (minutos)	Gratuidad	Ampliación de funcionalidades con versión de pago	Nº de Apps gratuitas	Nº de Apps permitidas previo pago	Posibilidad de publicación (implica pago)	Monetización	Publicidad en versiones gratuitas	Requerimientos técnicos: instalación de software, apps, etc	Autocompilación	Modificación de App una vez publicada	Almacenamiento de datos en el teléfono	Plantillas predefinidas	Posibilidad de creación desde cero
MIT app Inventor	Minimos	30	✓	✗	Ilimitadas	N/A	✗	✗	✗	✓	✓	✗	✓	✗	✓
Mobincube	✗	10	✓	✓	Ilimitadas	N/A	✓	✓	✓	✗	✓	✓	✗	✓	✓
Buzztouch	✗	60	✓	✗	3	50	✗	✓	✗	✓	✗	✓	✗	✓	✗

Tabla 3.2: Comparativa de características entre herramientas seleccionadas

- Tiempo de documentación previa para desarrollo de una app básica → Este tiempo es mayor con respecto a las otras dos herramientas.
- Número de aplicaciones gratuitas → Con Buzztouch el número de apps gratuitas se restringe a 3 mientras que con MIT App Inventor 2 y Mobincube se pueden crear un número ilimitado de aplicaciones.
- Autocompilación → Con Buzztouch habrá que descargar el código fuente del proyecto una vez generado y compilarlo usando eclipse, lo cual supone un tiempo adicional en el desarrollo de la aplicación y la realización de pruebas y modificaciones. Este es un aspecto importante a tener en cuenta dado que lo interesante de estas aplicaciones es el poco tiempo de desarrollo necesario, así como dar la posibilidad de realizar aplicaciones Android a personas que no tengan conocimientos de programación. Si tenemos en cuenta este aspecto, la mayoría de personas que no tienen conocimientos de programación, probablemente nunca hayan trabajado con eclipse lo cual supondrá un tiempo adicional de configuración y documentación sobre su uso.
- Posibilidad de creación desde cero → Con Buzztouch no es posible crear aplicaciones desde cero sino que habría que partir de plantillas previamente diseñadas.

Finalmente, concluimos con que las herramientas elegidas serían **MIT App Inventor** y **Mobincube**, que se adaptan perfectamente a los requerimientos de la aplicación a realizar. En los siguientes apartados se explicarán con detalle cada una de las herramientas seleccionadas.

3.1.1 MIT App Inventor 2

3.1.1.1 Introducción a MIT App Inventor 2

App Inventor es una aplicación que permite crear aplicaciones para sistemas operativos Android. Google Labs lanzó su primera versión de Google App Inventor en julio de 2010, con el objetivo de poder ofrecer una solución que permitiera crear nuevas aplicaciones de software a usuarios sin conocimientos previos de programación. En diciembre de 2011 Google anunció el fin de soporte de Google App Inventor, liberando el código y convirtiéndose en un proyecto Open Source, el cual fue continuado por el Massachusetts Institute of Technology (MIT), soportando la aplicación desde entonces, y lanzando mejoras que hacen que en la actualidad MIT App Inventor conjugue sencillez y potencia en una herramienta de desarrollo de software para uso particular y empresarial.

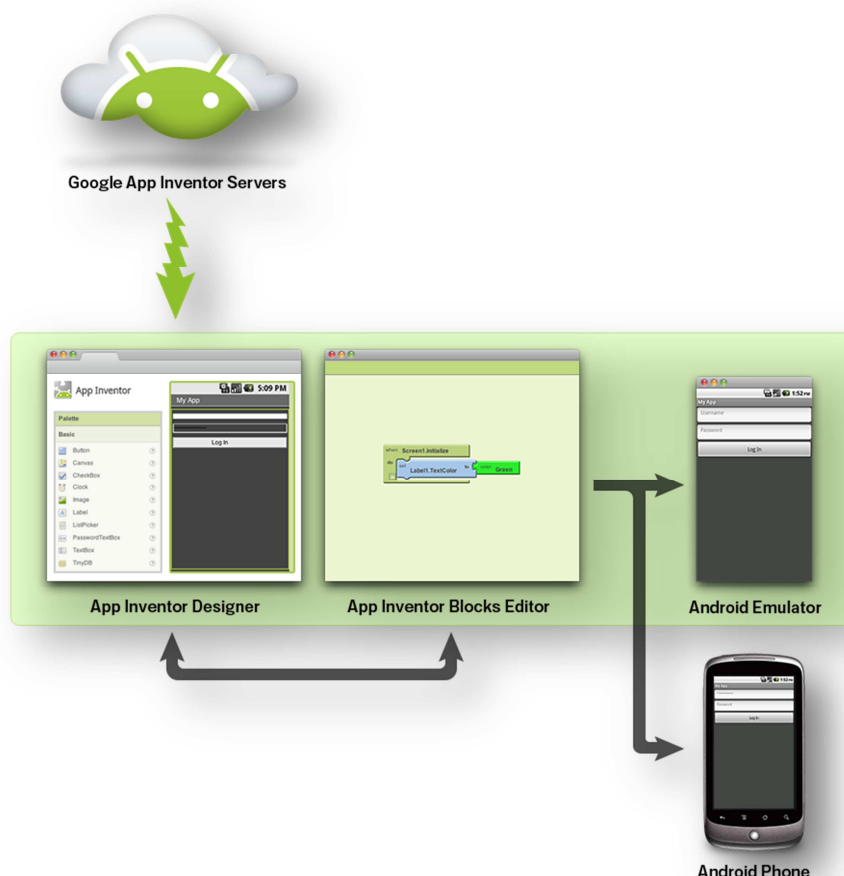


Ilustración 3.1: Diagrama de funcionamiento de App Inventor

3.1.1.2 Análisis de MIT App Inventor 2

➤ Instalación en Windows

La instalación de App Inventor en Windows consta de dos partes y unos requisitos previos.

Requisitos previos: Es necesario tener instalado Java 6 (también llamado Java 1.6)

Pasos de la instalación:

1. La instalación de App Inventor, cuyo proceso será igual para sistemas operativos Windows XP, Vista y 7.
2. La instalación de drivers para los dispositivos Android que queramos utilizar para nuestras pruebas, smartphones o tablets.

➤ Entorno de diseño

App Inventor Designer es el entorno de diseño donde se podrán seleccionar los objetos que serán utilizados en las aplicaciones, ubicarlos y darles el aspecto que se desee.

Para acceder dentro de App Inventor Designer, primero es necesario autenticarse, es decir acceder a: **<http://appinventor.mit.edu/>**, posteriormente elegir la opción **Create**, esta acción redirigirá a la página de Gmail, en donde se tendrá que introducir los siguientes datos: **Correo electrónico y contraseña** (previamente creado), y aparece la página principal:

El entorno de diseño consta de seis secciones (Cabecera, Palette, Viewer, Components, Media, Properties) y tiene el siguiente aspecto:

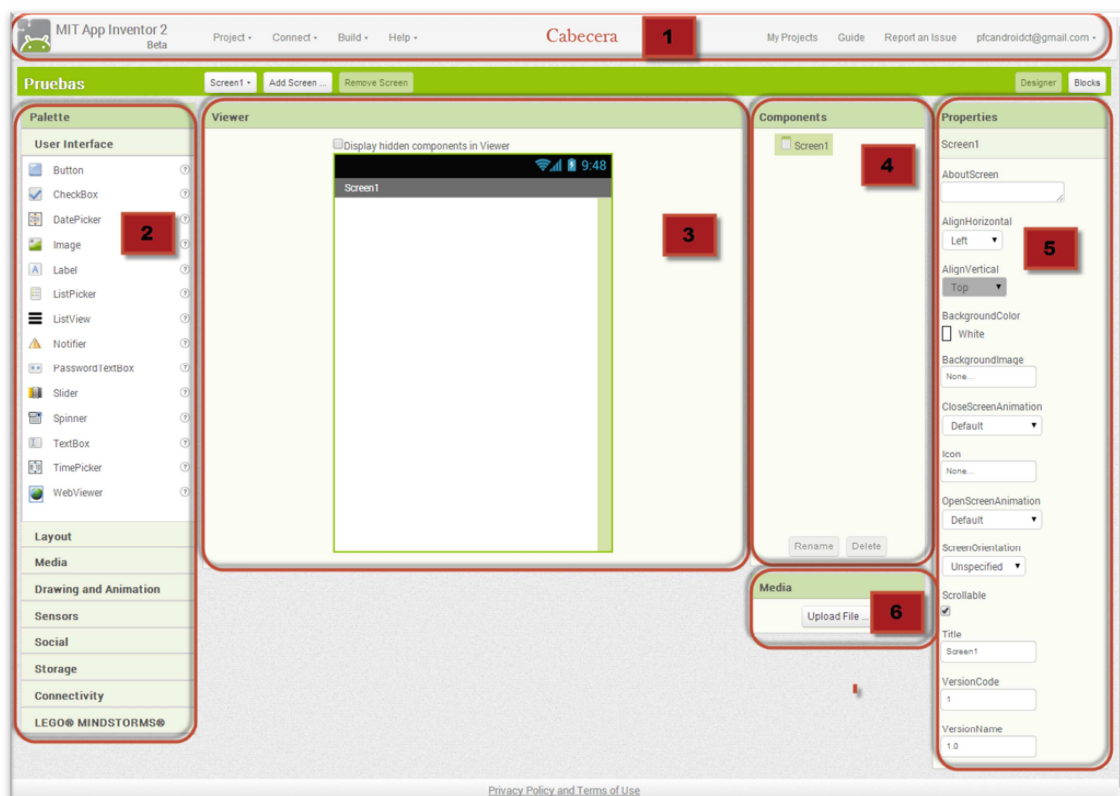


Ilustración 3.2: Entorno de diseño de App Inventor 2

1. Sección Cabecera:

- *My Projects* → opción con la que se podrá acceder al inventario de los proyectos que se tengan grabados.

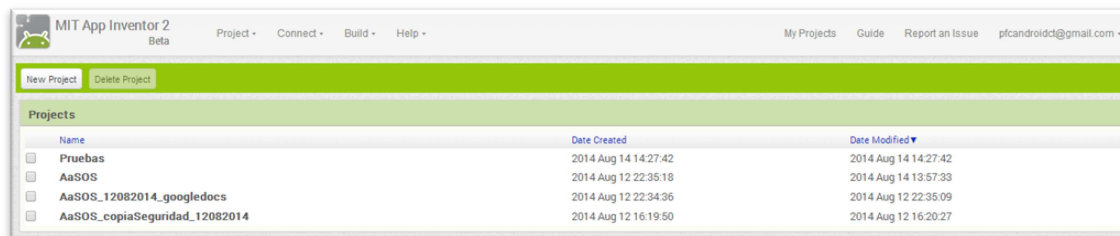


Ilustración 3.3: Inventario de proyectos con MIT App Inventor 2

- *New Project* → creación de nuevo proyecto.
- *Delete Project* → elimina proyecto.

2. Sección Palette

En la parte izquierda de la pantalla de diseño se encuentran todos los objetos que se podrán usar para nuestra aplicación, todos ellos separados dependiendo de sus características (User Interface, Layout, Media, Animation, etc).

Para insertar un objeto a la pantalla de diseño simplemente habrá que hacer clic sobre el objeto y arrastrar hasta la posición deseada, y así sucesivamente hasta que se termine el diseño.

3. Sección Viewer

En el centro de la pantalla de diseño se encuentra el visualizador de las pantallas que componen la aplicación.

Para añadir objetos de la Palette a una pantalla de la aplicación habrá que arrastrar el objeto seleccionado y soltarlo sobre la pantalla.

4. Sección Components

Esta ventana muestra todos los objetos, visibles y no visibles, que se encuentran en la pantalla que se está creando.

Existen 2 botones en la parte inferior de la ventana de Components:

- *Rename* → donde se podrá cambiar el nombre de un objeto, seleccionándolo previamente pinchando sobre él.
- *Delete* → para eliminar un objeto, seleccionándolo previamente pinchando sobre él.

Cuando se seleccione un objeto en la ventana de Components aparecerán sus propiedades modificables en la ventana de Properties.

5. Sección Properties

La ventana Properties permite modificar las propiedades de los objetos incluidos en la aplicación. Dependiendo del tipo de objeto las propiedades modificables variarán, siendo por ejemplo para un botón: color de fondo, las propiedades del texto (color, tamaño, apariencia, tipo de fuente), ancho, alto, posición, entre otras.

6. Sección Media

La ventana Media permite incluir a la aplicación ficheros de imágenes y sonido, los cuales podrán ser utilizados durante el desarrollo.

Los ficheros subidos mediante la ventana Media podrán ser utilizados en cualquier ventana de la aplicación. Seleccionando el archivo se dará la posibilidad de eliminarlo o descargarlo al equipo mediante las opciones *Delete* y *Download to my computer* respectivamente. Nota: se recomienda que el archivo sea menor a 5mb

➤ Editor de bloques

Para poder programar el comportamiento de una aplicación con éxito es necesario haber analizado previamente: Qué objetos tendrá la aplicación, qué aspecto tendrá cada objeto, cómo se comportará cada objeto ante determinados eventos. El análisis ayuda a afrontar la programación de una manera más sencilla.

Una vez insertados los objetos de la aplicación mediante el entorno Designer, desde App Inventor Blocks Editor se podrá definir su comportamiento.

App Inventor fue diseñado para que una persona sin conocimientos previos de programación pudiera crear sus propias aplicaciones en Android, por ese motivo la programación de los objetos de la aplicación se realizará de una manera muy sencilla e intuitiva mediante sistema de bloques.

Para ejecutar Blocks Editor es necesario pulsar el botón “Blocks” que se encuentra situado en la parte superior derecha de la ventana de diseño.

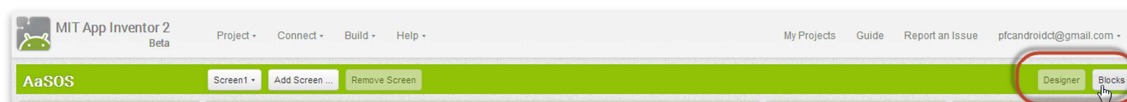


Ilustración 3.4: Selección de edición de bloques con MIT App Inventor 2

Al ejecutar Blocks Editor se muestran tres secciones diferentes, los cuales se indican en la siguiente imagen:

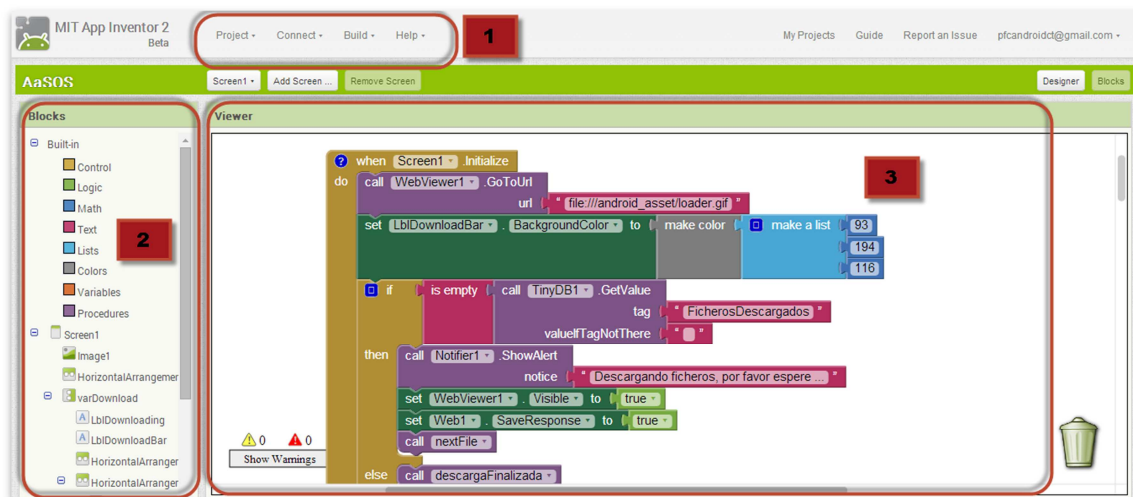


Ilustración 3.5: Editor de bloques con MIT App Inventor 2

1. *Sección Cabecera*

En la barra superior aparece la Sección Cabecera y tiene las opciones *Project – Connect – Build – Help*, con los que se podrá administrar los proyectos:

- *Project* → permite crear un nuevo proyecto, guardarlo, importarlo y exportarlo.

- *Connect* → permite conectar con código QR, emulador, USB o reiniciar la forma de conectarse.
- *Build* → construye la aplicación apk con código QR o construye el apk y lo descarga al PC.
- *Help* → contiene documentación de ayuda de appinventor, así como foros y tutoriales.

2. *Sección Blocks*

En el lateral izquierdo se encuentra la *Sección Blocks* que contiene tres carpetas:

- *Built-In* → donde se encuentran todas las herramientas y opciones que se pueden utilizar para programar las aplicaciones.
- *Screen* → aquí aparecerán los objetos que contiene la pantalla (screen) en desarrollo, pudiendo modificar el comportamiento de sus eventos y las propiedades.
- *Any Components* → en este apartado se podrán definir características comunes para los objetos insertados en la aplicación.

3. *Sección Viewer*

En la parte central de la pantalla se encuentra la *Sección Viewer*, que es un visualizador de los bloques que permiten definir el comportamiento de la aplicación.

Para añadir un objeto de Blocks, hay que hacer clic en el apartado de control, colores, matemáticas, texto o en cualquier objeto de Screen, aparecerán en el Viewer una serie de bloques, de esta lista se hará clic en el bloque que se desee añadir, después se arrastrará hasta el bloque seleccionado y se soltará sobre la pantalla del Viewer.

➤ *Generación de la App*

Desde la pestaña “Build” de la cabecera, se podrá descargar una Demo de la app en cualquier dispositivo. Se podrá generar un código QR para descargar la App en el dispositivo o guardar el .apk en el PC para posteriormente instalarlo en el dispositivo deseado.

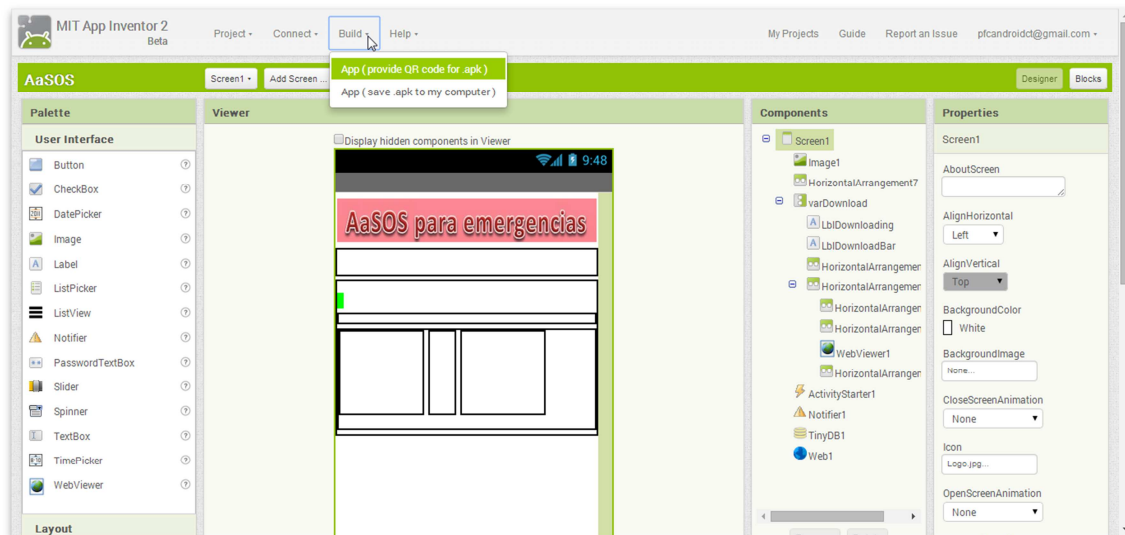


Ilustración 3.6: Compilación con MIT App Inventor 2

Si la opción elegida es la de obtener el código QR, en unos instantes se generará y mostrará por pantalla dicho código.

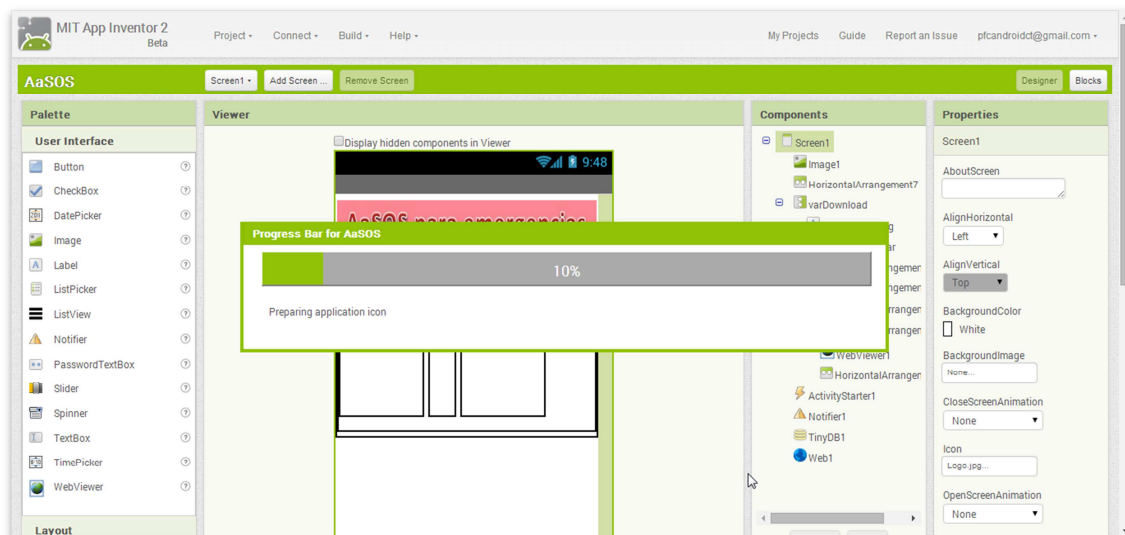


Ilustración 3.7: Progreso de compilación con MIT App Inventor 2

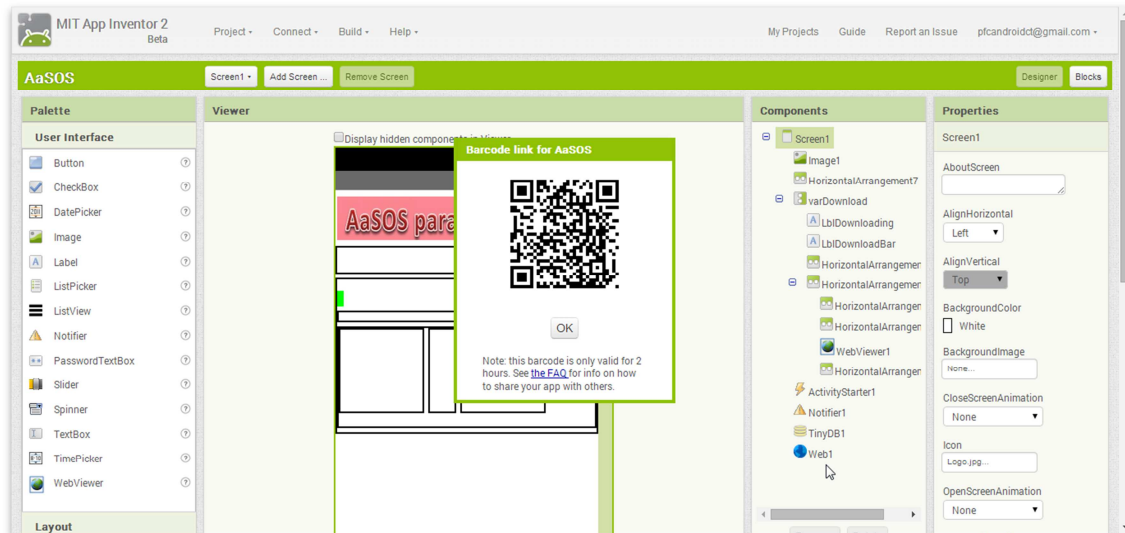


Ilustración 3.8: Descarga de .apk con lectura de código QR en MIT App Inventor 2

Para leer este código QR desde el dispositivo móvil, es necesario haberse descargado desde Play Store e instalado la aplicación MIT AI2 Companion . Una vez instalada, se abrirá y leerá el código QR para proceder a la descarga e instalación de la aplicación.

➤ Publicación de la App

MIT App Inventor 2 aún no permite publicar directamente las aplicaciones en las tiendas Google Play, Apple Store y Windows Phone. Es necesario realizar una transformación del .apk generado para poder publicar la aplicación.

3.1.2 Mobincube

3.1.2.1 Introducción a Mobincube

Mobincube es una plataforma que permite desarrollar aplicaciones móviles a cualquier persona, aunque no tenga ningún tipo de conocimientos en programación, de forma sencilla y rápida. Actualmente permite crear Apps nativas para las principales plataformas móviles de la actualidad: iOS (iPhone, iPad, iPod), Android (smartphones y tablets), Windows Phone y HTML5.

Mobincube se ofrece con una solución web, por lo que no precisa instalar ningún software previamente. Ofrece un servicio muy completo con la ayuda de asistentes, una gran lista de plantillas y un entorno de trabajo visual muy intuitivo y funcional.

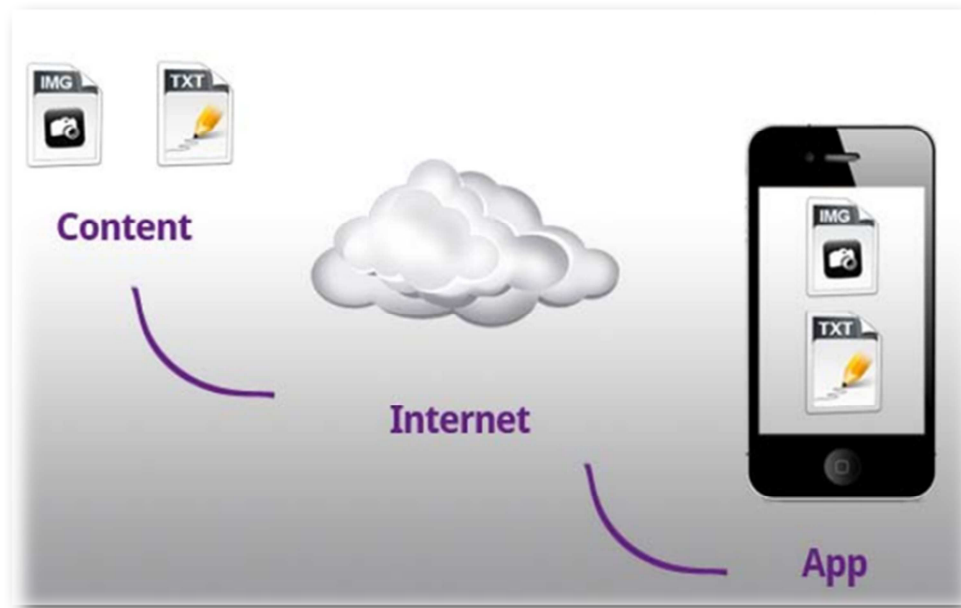


Ilustración 3.9: Diagrama general Mobincube

3.1.2.2 Análisis de Mobincube

➤ Instalación en Windows

No requiere instalación previa al tratarse de una solución web. La url de acceso a mobincube es <http://www.mobincube.com/>.

Para empezar a trabajar con mobincube será necesario registrarse de forma gratuita en la web completando los datos que se muestran en la siguiente imagen:

La imagen muestra la interfaz de usuario de la página de registro de Mobincube. En la parte superior izquierda, el logo de Mobincube con el tagline 'The App builder for everyone!'. A la derecha, hay enlaces para 'Tour', 'Pricing', 'Make money' y 'Support', junto con botones para 'Free Sign up' y 'Login'. El formulario de registro contiene cuatro campos de entrada: 'Insert username', 'Insert password', 'Repeat password' y 'E-mail'. Debajo de estos campos, hay un checkbox con el texto 'I've read and agree with the Terms and conditions'. Un botón rojo con el texto 'Create account' está situado debajo del checkbox. En la parte inferior, hay un texto pequeño que dice: '*Once you've registered, you will receive an email to validate this Mobincube demo account.'

Ilustración 3.10: Página de registro en Mobincube

➤ Entorno de diseño

○ *Creación de una nueva App:*

Para crear una nueva app en Mobincube es tan sencillo como situarse en la parte superior izquierda del panel “Mis Apps” y hacer click sobre el icono “+”.

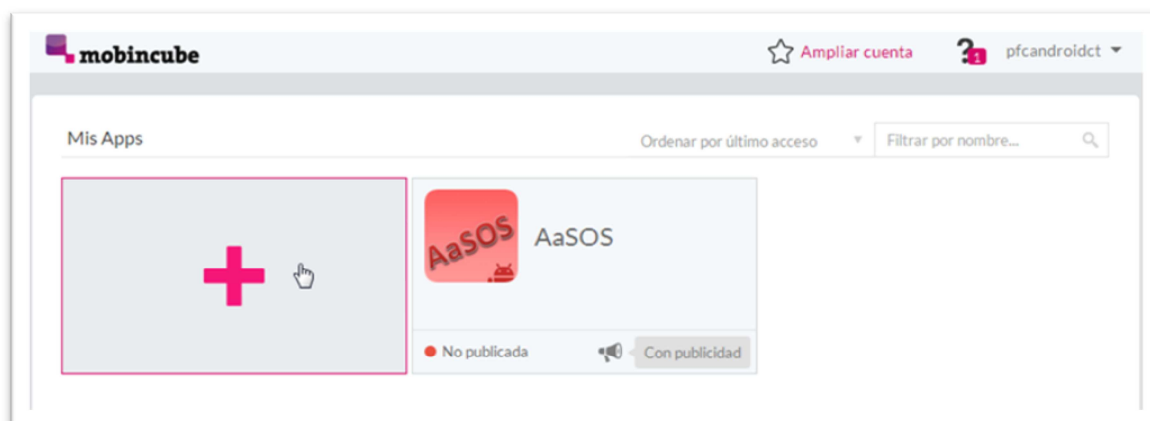


Ilustración 3.11: Creación nueva app con Mobincube

En la siguiente pantalla emergente se deberá introducir un nombre para la app y hacer clic en el botón “Crear”.

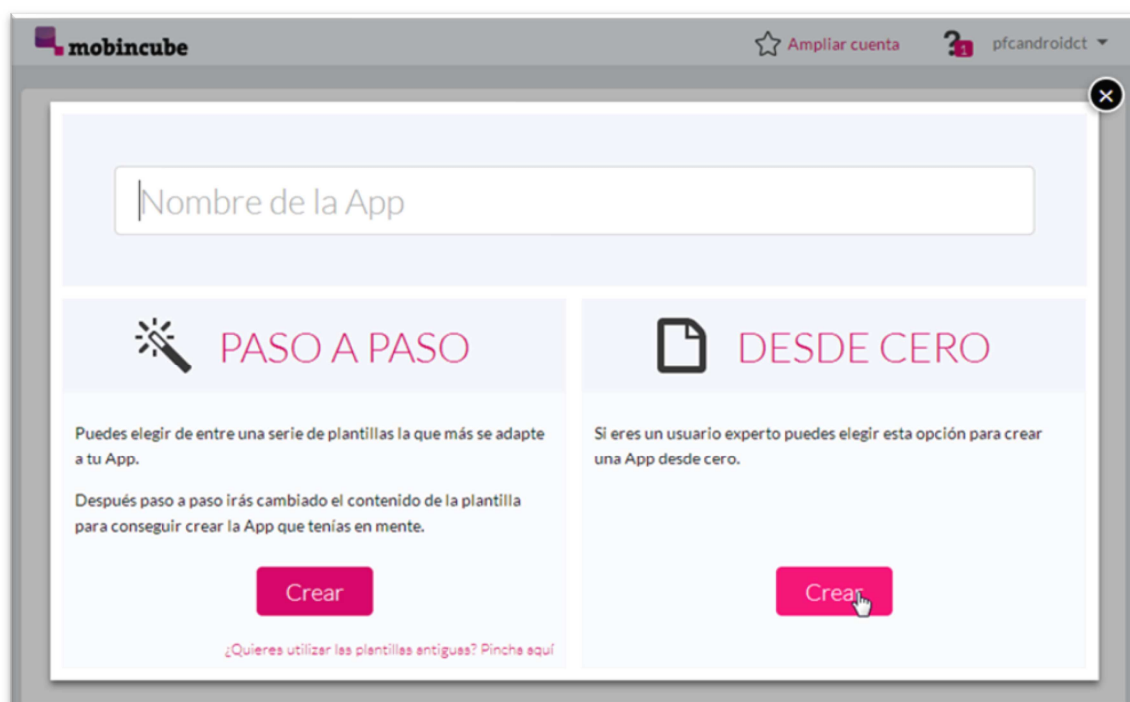


Ilustración 3.12: Insertar nombre app con Mobincube

Existen 2 botones “crear” permitiendo los siguientes modelos de creación de App:

- Crear una App paso a paso mediante el uso de plantillas.

Al pulsar el botón “crear” se mostrará la siguiente pantalla de selección de plantilla para comenzar con el desarrollo de la app a partir de ella.

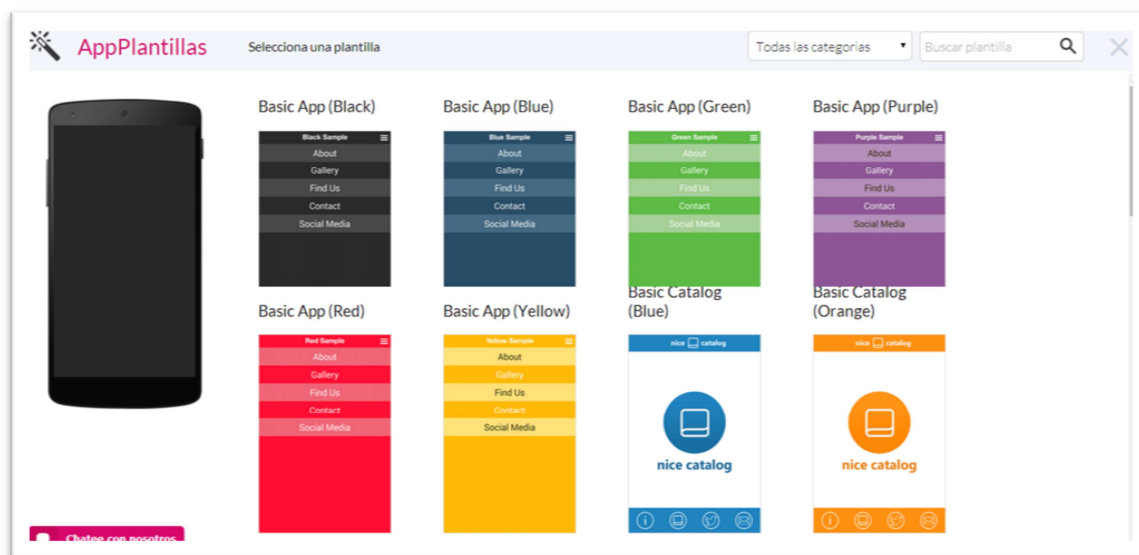


Ilustración 3.13: Selección de plantilla con Mobincube

- Crear una App desde cero.

Dado que el método de crear una App desde cero es el usado en el presente proyecto final de carrera, se explicará con más detalle en los siguientes apartados.

- *Modificación de datos*

Haciendo click en la pestaña “Info” de la barra de navegación superior de la pantalla, se mostrará el Panel de datos de la app que permitirá editar la información acerca de la misma (nombre, descripción, categoría, idioma, notas de la versión, icono)

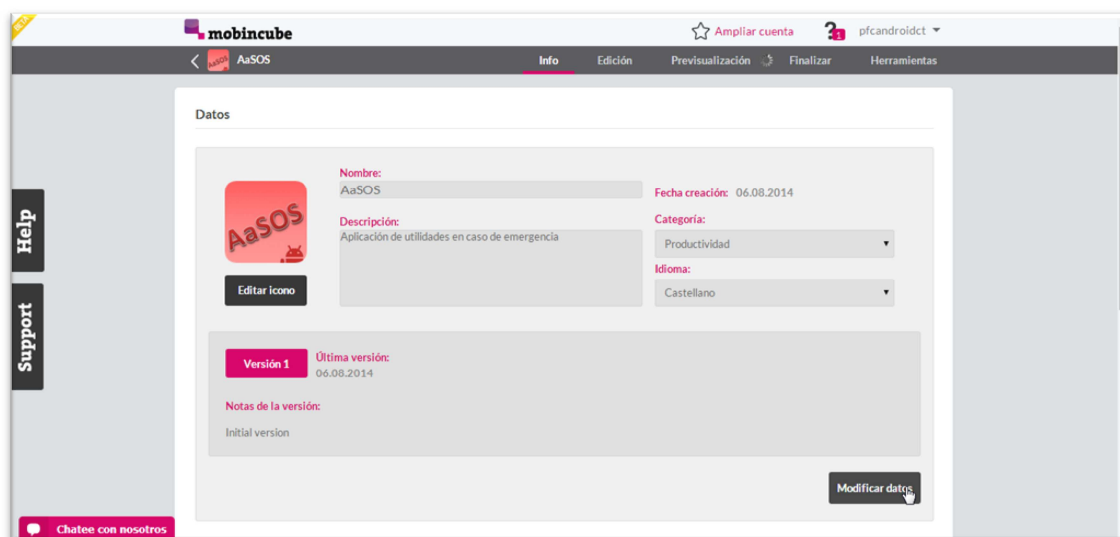


Ilustración 3.14: Edición de información de la app con Mobincube

○ *Creación de la estructura*

La creación de la estructura de la app se realizará desde la pestaña “Edición”. A continuación se muestra en la siguiente imagen el aspecto general que presenta la interfaz de Mobincube al crear una nueva aplicación.

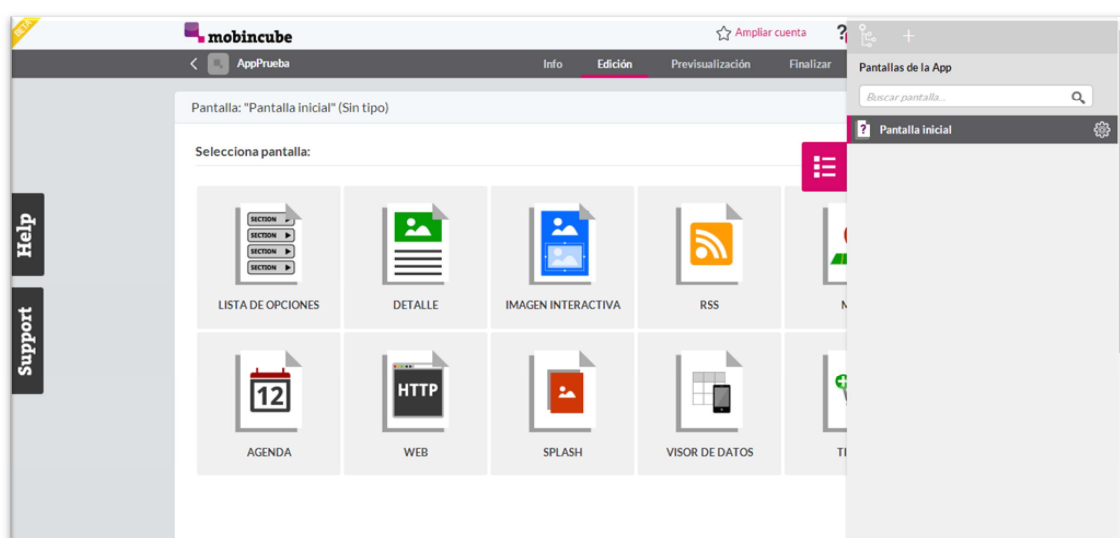


Ilustración 3.15: Selección de pantalla con Mobincube

Al crear una nueva aplicación, se tendrá que seleccionar el tipo de pantalla que se desea crear para la pantalla principal.

Mobincube dispone de 10 tipos de pantallas con los que se podrá crear una completa aplicación, añadiendo diferentes opciones y acciones para el buen funcionamiento de la app. Los 10 tipos de pantalla son:

- Lista de opciones

- Detalle
- Imagen interactiva
- RSS
- Mapa
- Agenda
- Web
- Splash
- Visor de datos
- Tienda

Cada tipo de pantalla aparece representado por su respectivo icono y únicamente será necesario hacer click sobre el tipo deseado para empezar con el desarrollo de la app.

La pantalla principal es la primera que se carga al abrir una App. Desde ella se puede navegar a los distintos apartados de una App. Las diversas pantallas de una app se vinculan directa o indirectamente con la pantalla principal formando así el esquema de navegación de la App, el cual se puede ver a la derecha de la pantalla. Este esquema de navegación se puede ver representado a modo árbol, permitiendo observar la interconexión de las páginas obteniendo un esquema de la navegabilidad, o como listado de las pantallas existentes.

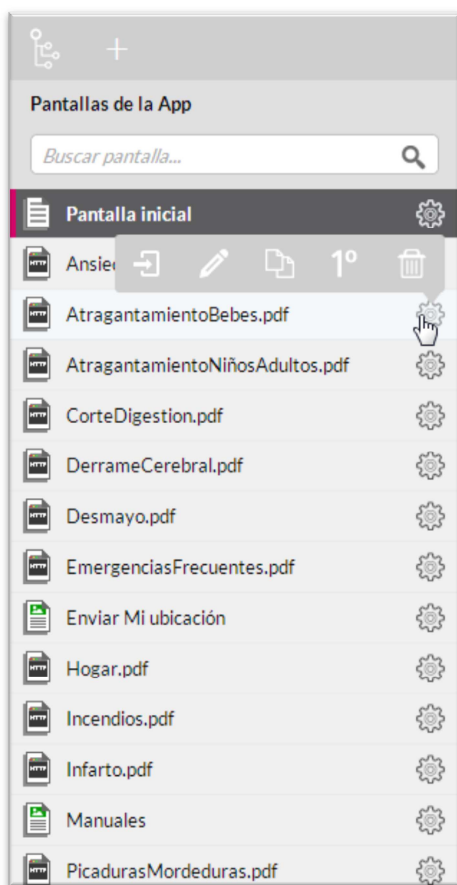


Ilustración 3.16: Listado de pantallas con Mobincube

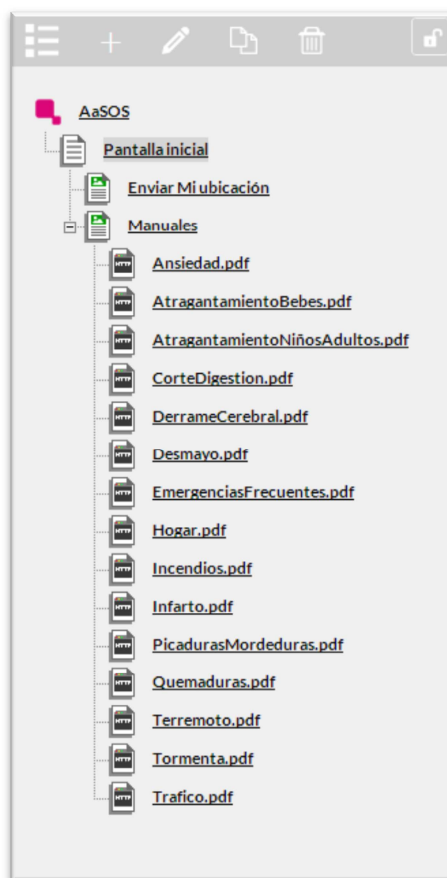


Ilustración 3.17: Esquema de pantallas en modo árbol con Mobincube

Se podrán realizar las siguientes acciones a partir del árbol de pantallas que componen la aplicación:



Ilustración 3.18: Acciones permitidas sobre una pantalla con Mobincube

1. Añadir nueva pantalla
2. Abrir pantalla
3. Renombrar pantalla
4. Duplicar pantalla
5. Establecer pantalla como pantalla principal
6. Borrar pantalla

○ *Creación de contenidos*

Al igual que la creación de la estructura, la creación de contenidos de la app también se realizará desde la pestaña “Edición” de la barra de navegación superior de la pantalla.

Para la creación de contenidos Mobincube cuenta con una barra de herramientas a la derecha de la pantalla tal y como se muestra en la siguiente imagen:

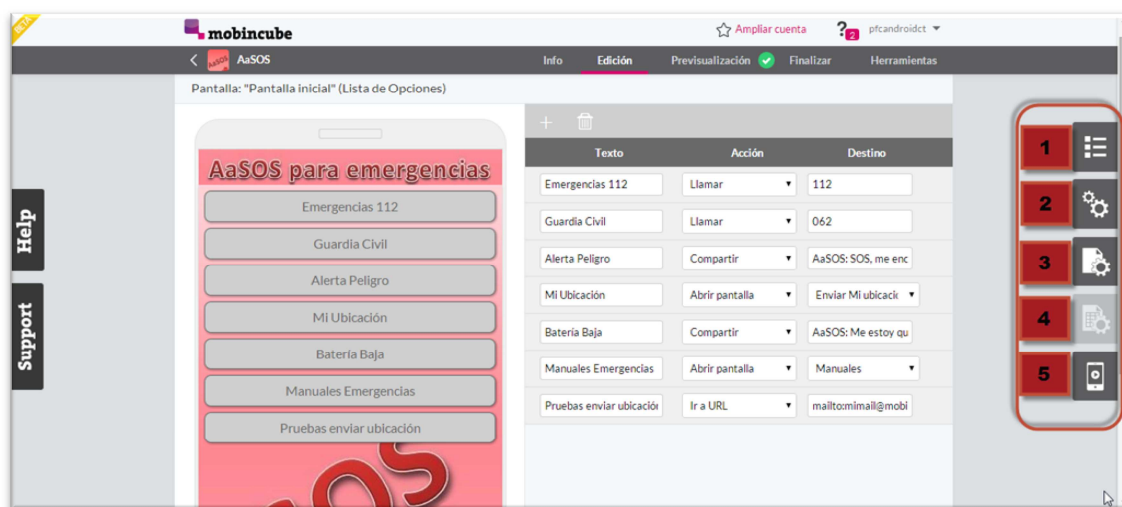


Ilustración 3.19: Barra de herramientas de Mobincube

La barra de herramientas cuenta con las siguientes opciones:

1. *Árbol de pantallas*

Esta opción ya se ha detallado en el apartado anterior de “*Creación de la Estructura*”

2. *Propiedades de la aplicación*

Dentro de esta pestaña de la barra de herramientas se podrán editar las propiedades generales de la aplicación. Estas son las que afectan a la aplicación en general y sobre ellas prevalecen siempre las propiedades de una pantalla en particular. Las propiedades que se podrán modificar son: Imagen de cabecera, colores del entorno, fuente del texto, colores de la barra de comandos, diseño de la “Lista de opciones”, diseño de la Agenda, diseño de los RSS.



Ilustración 3.20: Propiedades generales de la aplicación con Mobincube

3. *Propiedades de la pantalla*

Esta opción de la barra de herramientas permite establecer diferentes propiedades para cada una de las pantallas de la aplicación. Por tanto, esta cambia dependiendo de la pantalla con la que se esté trabajado en cada momento.

No obstante, existen propiedades comunes para todos los tipos de pantallas como son: imagen de cabecera, fondo de pantalla, comandos y temporizador. Las opciones que se editan desde las propiedades de pantalla prevalecen sobre las fijadas desde propiedades de la aplicación.



Ilustración 3.21: Propiedades comunes para todos los tipos de pantallas con Mobincube

4. *Propiedades de la vista*

Esta opción únicamente estará disponible para las pantallas de tipo “Visor de Datos” ofreciendo la posibilidad de buscar dentro de los campos de texto de una base de datos y de acceder al gestor de barras.

La característica del buscador se puede activar en las vistas tipo lista, y se activa y configura en el panel de la derecha pulsando el icono de la lupa.



Ilustración 3.22: Propiedades de la vista con Mobincube

5. *Simulador*

El simulador facilita la visualización del trabajo realizado dando una visión estática pero muy útil de la pantalla actual. Permite navegar por toda la aplicación siempre y cuando se comience la simulación desde la pantalla principal de la App.

Esta simulación intenta ser lo más fiel posible a todas las versiones de la app ya que Mobincube genera versiones para diferentes plataformas y dispositivos: iPhone, iPad, Android, etc. Por otro lado, también permite ejecutar el simulador en modo tableta.



Ilustración 3.23: Simulador con Mobincube

➤ Generación de la App

Desde la pestaña “Previsualización” de la barra de navegación superior de la pantalla, se podrá descargar una Demo de la app en cualquier dispositivo. Simplemente habrá que hacer click en el botón “solicitar” y en unos minutos se generará el enlace e imagen QR para proceder con la descarga.

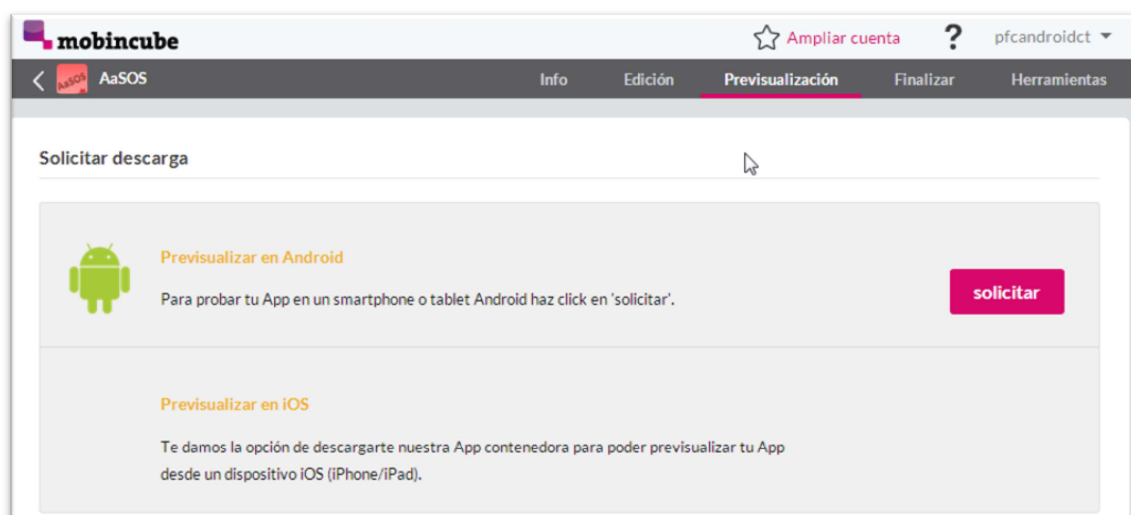


Ilustración 3.24: Compilación con Mobincube

Mientras se está generando la Demo se puede continuar editando la App.

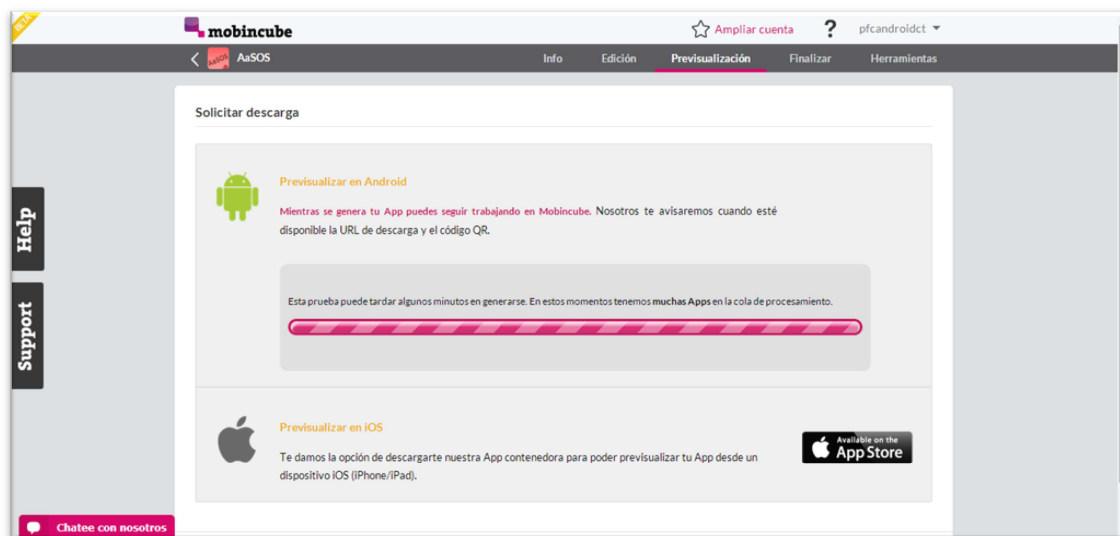


Ilustración 3.25: Progreso de compilación con Mobincube

Por último insertando la URL o capturando la imagen QR con el dispositivo móvil, se procederá a la descarga de la app.

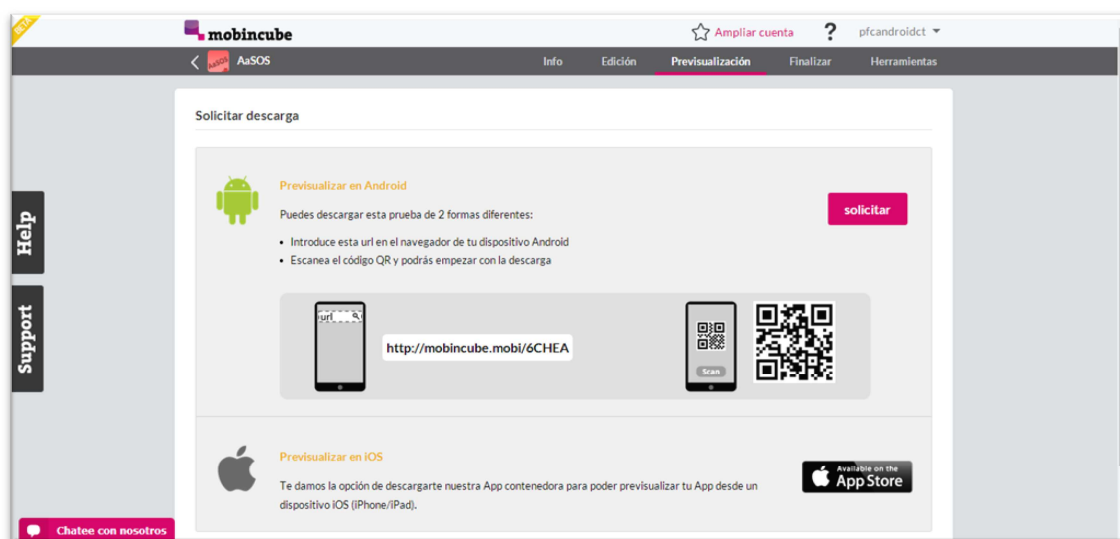


Ilustración 3.26: Descarga de .apk con lectura de código QR en Mobincube

➤ Publicación de la App

La publicación de la App con Mobincube es algo tan sencillo como hacer click en la pestaña “Finalizar” de la barra de navegación superior de la pantalla y se redirigirá a la zona de publicaciones. Desde ahí se podrá elegir dónde y cómo se desea publicar la App.

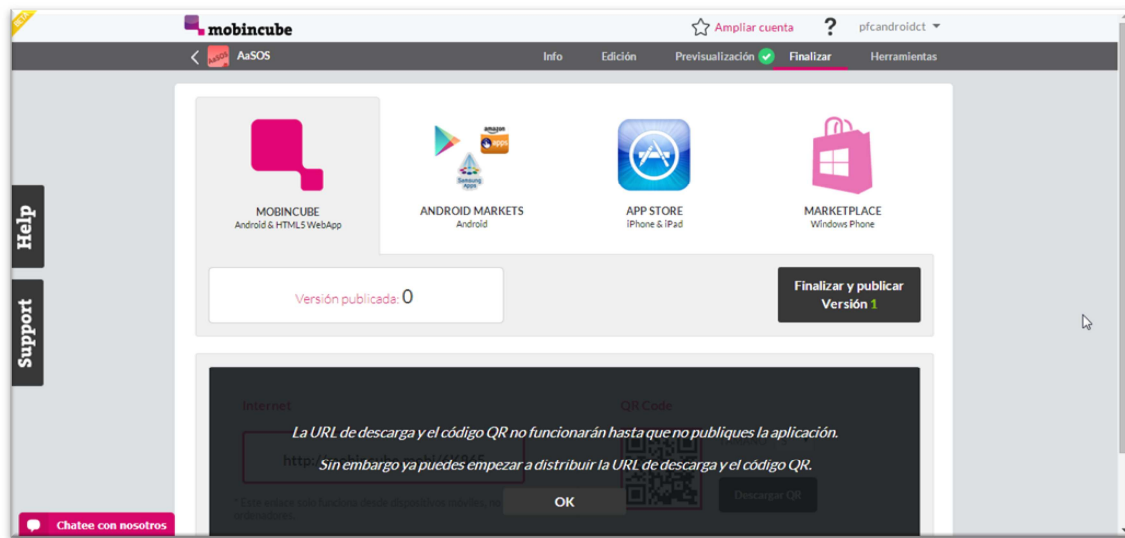


Ilustración 3.27: Publicación de la aplicación con Mobincube

3.2 Comparativa entre MIT App Inventor 2 y Mobincube

A continuación se va a realizar una comparativa de tiempos, fortalezas, debilidades y otros puntos de interés entre las herramientas MIT App Inventor 2 y Mobincube. Es importante tener en cuenta que los tiempos indicados de realización no tienen en cuenta la generación de imágenes, logo, documentación y búsqueda de información necesaria para añadir a la aplicación. En cuanto al tiempo de documentación indicado en la tabla, únicamente se refiere a lo relacionado con las necesidades de la aplicación desarrollada.

	MIT APP INVENTOR 2	MOBINCUBE
<u>Tiempo de documentación/estudio/suporte</u>	10 horas	5 horas
<u>Tiempo de realización</u>	6 horas	3 horas
<u>Conocimientos previos</u>	Se necesitan unos conocimientos mínimos de programación	No se necesitan conocimientos previos de programación
<u>Puntos fuertes</u>	<ul style="list-style-type: none"> ✓ Aporta autonomía a los que no tienen la destreza y conocimientos de los desarrolladores web o profesionales. Gracias a esta herramienta, cualquiera puede crear su propia 	<ul style="list-style-type: none"> ✓ Simplicidad ✓ No es necesario instalar un IDE ✓ Permite realizar de una forma muy sencilla la publicación de las aplicaciones en las

	<p>aplicación, lo que facilita el acceso a la programación sin ser un especialista en informática.</p> <ul style="list-style-type: none"> ✓ No es necesario instalar un IDE ✓ Desarrollo rápido de aplicaciones con bajos niveles de error. ✓ Almacenamiento en la nube. 	<p>tiendas Google Play, Apple Store y Windows Phone.</p> <ul style="list-style-type: none"> ✓ Almacenamiento en la nube.
<u>Puntos débiles</u>	<ul style="list-style-type: none"> ✓ Precisamente uno de sus puntos fuertes da lugar a uno de sus puntos débiles que es la calidad en las aplicaciones. El riesgo es que este servicio puesto a disposición de todos, disminuye la calidad global de la oferta de las aplicaciones existentes para Android. ✓ Aún no permite publicar las aplicaciones en las tiendas Google Play, Apple Store y Windows Phone. 	<ul style="list-style-type: none"> ✓ Presentación de publicidad para la versión gratuita. ✓ No permite almacenar y recuperar datos en una BBDD local, sino que es necesario usar un servidor externo para la versión gratuita.
<u>Visualización de documentos almacenados en la memoria SD del dispositivo</u>	Si	No

<u>Manejabilidad</u>	<ul style="list-style-type: none"> ✓ Se ejecuta vía web, sólo es necesario ejecutar una pequeña aplicación para mostrar el Editor de Bloques (Block Editor) y el emulador. ✓ Permite conectarse directamente al equipo Android, aunque en algunos casos no será posible, por ello incluye un emulador para probar las aplicaciones que se desarrollen. ✓ No se necesita ningún tipo de compilador sino que todo se realiza en el entorno de App Inventor 	<ul style="list-style-type: none"> ✓ Se ejecuta vía web sin necesidad de llevar a cabo ninguna instalación. ✓ Incluye un simulador para probar las aplicaciones vía web sin necesidad de instalarla en el dispositivo. ✓ Entorno de trabajo visual y con la ayuda de asistentes. ✓ No se necesita ningún tipo de compilador sino que todo se realiza en el entorno de Mobincube.
<u>Límite de complejidad de una aplicación</u>	Este tipo de herramientas son sobre todo muy útiles para las pequeñas empresas que tienen necesidades modestas y no precisan de aplicaciones complejas y pesadas en volumen de datos. De hecho, este tipo de herramientas no permiten crear aplicaciones a medida. No obstante, no permite aplicaciones complejas pero si completas	
<u>Tipos de aplicaciones</u>	App Inventor permite crear cualquier tipo de aplicaciones desde juegos a sitios web o desarrollar simples aplicaciones mediante compilación de texto e imágenes en un feed RSS útiles para pequeñas empresas o negocios.	Es posible crear diferentes clases de aplicaciones pues dispone de un catálogo de plantillas prediseñadas en las que se puede inspirar el desarrollador, como presentaciones de productos, guías de viajes, eBooks, etc.
<u>Gratuidad del servicio</u>	Si	Mobincube es un software gratuito que permite crear apps ilimitadas aunque la

		<p>versión gratuita incluye tanto una pantalla inicial de Mobincube como anuncios. Es posible suscribirse a planes mensuales, muy económicos, si se quiere almacenar las apps en Mobincube para futuras actualizaciones y para disfrutar de servicios Premium adicionales. Además</p>
<p><u>Posibilidad de añadir publicidad para ganar dinero(Monetización)</u></p>	<p>No. Es posible añadir publicidad a la aplicación a través de empresas que se dedican precisamente a la monetización de las aplicaciones como pueden ser: Mobpartner, Admob o Leadbolt</p>	<p>Sí. Cuenta con un interesante servicio de monetización que permite gestionar de forma sencilla la integración de anuncios en la aplicación para ganar algo de dinero.</p>

Tabla 3.3: Comparativa entre MIT App Inventor y Mobincube

Capítulo 4

Caso Práctico - Desarrollo de Aplicación AaSOS.

4.1 Funcionalidad y objetivos de la aplicación AaSOS



Ilustración 4.1: Logo de la aplicación AaSOS

- *Nombre de la aplicación:* AaSOS
- *Descripción:* Se trata de una aplicación de utilidad en situaciones de emergencias o peligro.
- *Utilidades de AaSOS:*
 - ✓ Realización de llamada al número de emergencias 112
 - ✓ Realización de llamada al teléfono de la Guardia Civil 062.
 - ✓ Compartir ubicación actual del dispositivo móvil
 - ✓ Compartir alerta de peligro
 - ✓ Compartir aviso de batería baja
 - ✓ Visualización de manuales de prevención y primeros auxilios ante situaciones de emergencia.

El nombre que se le ha asignado a la aplicación ha sido pensado teniendo en cuenta la utilidad de la aplicación y la iniciativa de utilización de "Aa (Avisar a)" que fue propuesta en España en julio de 2009, conjuntamente por la Cruz Roja y el Ministerio del Interior español. Esta iniciativa fue propuesta con el fin de facilitar a los servicios que atiendan a una persona que sufra una emergencia, la localización de forma sencilla de los familiares o amigos de la misma, que de esta forma podrán acudir rápidamente al lugar donde se encuentra el afectado o dar instrucciones sobre aspectos importante de la víctima, como alergias, posible enfermedad o medicación especial que pueda estar tomando.

Por tanto, el nombre de esta aplicación por un lado aporta información sobre su utilidad ante situaciones de emergencia, y por otro hará que aparezca de las primeras en la lista de aplicaciones ordenadas alfabéticamente.

4.2 Aplicación AaSOS

4.2.1 Pantalla principal

Al acceder a la aplicación, en la pantalla principal se podrá ver un menú de botones, con las acciones que se podrán realizar.



Ilustración 4.2: Pantalla principal AaSOS

4.2.2 Opciones de menú

➤ Emergencias 112

Al pulsar esta opción del menú, se realizará una llamada al número de emergencias 112.



Ilustración 4.3: Llamada 112 con AaSOS

➤ Guardia Civil

Al pulsar esta opción del menú, se realizará una llamada al número de la guardia civil 062. Esta llamada se realizará directamente sin pasar por la pantalla de marcación de número.



Ilustración 4.4: Llamada Guardia Civil con AaSOS

➤ Mi Ubicación

Al pulsar esta opción del menú, se compartirá la ubicación actual donde se encuentra el dispositivo móvil. Esta ubicación se podrá compartir haciendo uso de cualquiera de las aplicaciones instaladas que permitan compartir texto con otros dispositivos (correo, SMS, Facebook, WhatsApp...).

En primer lugar se mostrará una pantalla para seleccionar la aplicación que permitirá compartir la ubicación y una vez elegida se enviará la ubicación actual haciendo uso de dicha aplicación.

El texto que se enviará es el siguiente: *AaSOS: Mi ubicación actual en formato “latitud, longitud” es: xxxxxx, xxxxxx*



Ilustración 4.5.1: Envío de ubicación – Paso 1



Ilustración 4.5.2: Envío de ubicación – Paso 2



Ilustración 4.5.3: Envío de ubicación – Paso 3

➤ Alerta Peligro

Al pulsar esta opción del menú, se enviará una alerta solicitando ayuda en caso de encontrarse ante una situación de peligro. Se podrá compartir un mensaje de solicitud de ayuda haciendo uso de cualquiera de las aplicaciones instaladas que permitan compartir texto con otros dispositivos (correo, SMS, Facebook, WhatsApp...).

En primer lugar se mostrará una pantalla para seleccionar la aplicación que permitirá compartir el mensaje y una vez elegida se abrirá dicha aplicación para enviar el mensaje.

El texto que se enviará es el siguiente: *AaSOS: SOS, me encuentro en situación de peligro. Por favor, necesito ayuda.*

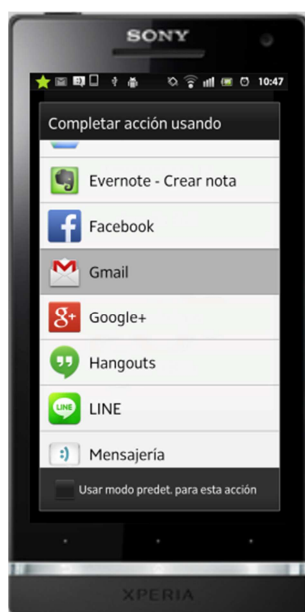


Ilustración 4.6.1: Enviar aviso peligro – Paso 1



Ilustración 4.6.2: Enviar aviso peligro – Paso 2

➤ Batería Baja

Al pulsar esta opción del menú, se enviará un mensaje indicando que el dispositivo se está quedando sin batería y que en breve se apagará. Se podrá compartir el mensaje de aviso haciendo uso de cualquiera de las aplicaciones instaladas que permitan compartir texto con otros dispositivos (correo, SMS, Facebook, WhatsApp...).

En primer lugar se mostrará una pantalla para seleccionar la aplicación que permitirá compartir el mensaje y una vez elegida se abrirá dicha aplicación para enviar el mensaje.

El texto que se enviará es el siguiente: *AaSOS: Me estoy quedando sin batería y en breve se me va a apagar el teléfono.*

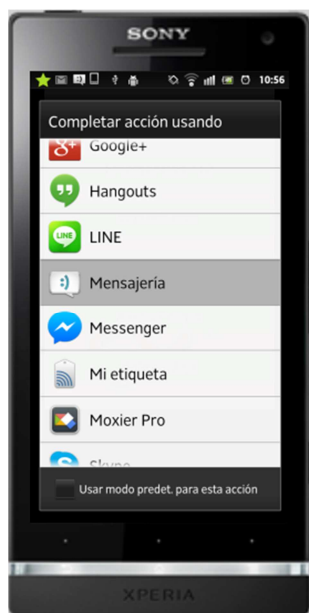


Ilustración 4.7.1: Enviar aviso batería baja – Paso 1



Ilustración 4.7.2: Enviar aviso batería baja – Paso 2

➤ Manuales Emergencias

Al pulsar esta opción del menú, se mostrará un listado de manuales de prevención y primeros auxilios ante determinadas situaciones de emergencia. Al hacer click sobre alguno de ellos, se abrirá el manual en modo lectura.

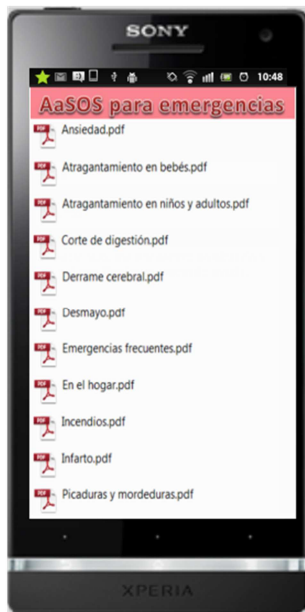


Ilustración 4.8.1: Visualización de manuales – Paso 1

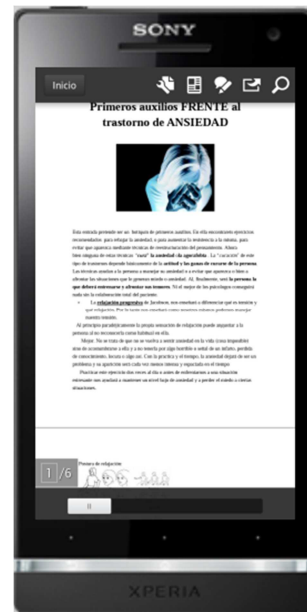


Ilustración 4.8.2: Visualización de manuales – Paso 2

4.2.3 Diferencias de AaSOS con MIT App Inventor y Mobincube

A la hora de realizar la misma aplicación con MIT App Inventor y Mobincube, cabe destacar que existen 2 diferencias claramente notables entre ellas. Por un lado tenemos el tema de la publicidad en Mobincube y por otro el poder almacenar Pdf's en la tarjeta SD del dispositivo móvil.

➤ Publicidad

Con Mobincube se mostrará antes de abrir la pantalla principal, una ventana de publicidad de Mobincube y durante la ejecución de la aplicación también irán apareciendo ventanas de publicidad. Además cuenta con un banner de anuncios que se mostrará en todo momento.

➤ Almacenar Pdf's en tarjeta SD

Con MIT App Inventor es posible almacenar documentos Pdf's en la tarjeta SD del dispositivo, lo cual supone una ventaja a la hora de querer visualizar los manuales de prevención y primeros auxilios en casos de emergencias, permitiendo visualizarlos de forma más rápida y sin necesidad de tener conexión a internet en ese momento. Esto no es posible con Mobincube que se ha tenido que optar por la opción de subir los ficheros a la web de "ISSU".

4.2.3.1 Aplicación AaSOS con MIT App Inventor 2

- URL: <http://appinventor.mit.edu/explore/>
- Problemas encontrados durante el desarrollo:

El principal problema encontrado durante el desarrollo con MIT App Inventor ha sido a la hora de descargar un conjunto de ficheros Pdf's a la memoria SD del dispositivo para tener acceso a ellos en cualquier momento durante la ejecución de la aplicación, sin necesidad de tener conexión a internet. Aunque esta herramienta tiene una sección llamada “media” destinada a subir ficheros para su uso en la aplicación a desarrollar, no es posible subir ficheros con extensión .Pdf para poder visualizarlos con algún visor de Pdf's una vez ejecutada la aplicación.

Se tuvo que optar entonces por la opción de subir los Pdf's a una cuenta de Google Drive y cuando la aplicación se ejecutase por primera vez, se procedería a la descarga de dichos documentos.

➤ Pantalla Principal

La primera vez que se ejecuta la aplicación después de haberla instalado, se realiza la descarga a la tarjeta SD del dispositivo de unos manuales de uso en caso de emergencias, para que puedan ser utilizados en cualquier momento sin necesidad de tener conexión a internet. Una vez finalizada la descarga se mostrará la pantalla inicial con el menú de opciones permitidas.



Ilustración 4.9.1: Descarga de manuales en proceso



Ilustración 4.9.2: Descarga de manuales finalizada

➤ Manuales Emergencias

Al hacer click sobre alguno de ellos, se abrirá el Pdf almacenado inicialmente en la tarjeta SD del dispositivo y se visualizará con algún visor de documentos Pdf's que se tenga instalado.

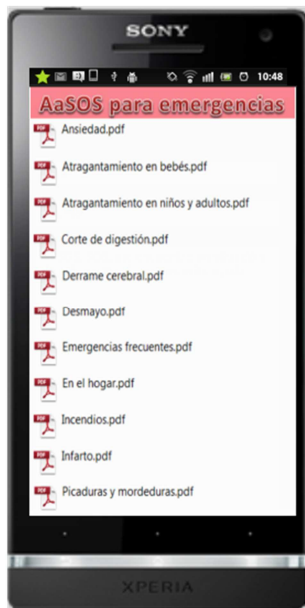


Ilustración 4.10.1: Visualización de manuales con MIT App Inventor 2 – Paso 1



Ilustración 4.10.2: Visualización de manuales con MIT App Inventor 2 – Paso 2

4.2.3.2 Aplicación AaSOS con Mobincube

- URL: <http://www.mobincube.com/>
- Problemas encontrados durante el desarrollo:

Se encontraron 2 problemas durante el desarrollo, que te tuvieron que tratar con el equipo de soporte de Mobincube:

- El primero de ellos es relacionado con la visualización de Pdf's. Con Mobincube no era posible almacenar Pdf's en el dispositivo para después acceder a ellos, por lo que se tuvo que optar por la opción de visualizar los documentos haciendo uso de un sistema como <http://issuu.com/> que permitiera mostrar el documento embebido en nuestra aplicación.

Issuu es un servicio en línea que permite la visualización de material digitalizado electrónicamente, como libros, portafolios, números de revistas, periódicos, y otros medios impresos de forma realística y personalizable. El servicio que ofrece Issuu en el área de publicación se puede comparar con lo que [Flickr](#) hace para compartir fotografías, y lo que [YouTube](#) hace para compartir video. El material subido al sitio es visto a través de un [navegador web](#) y está hecho para parecerse lo más posible a una publicación impresa, con un formato que permite la visualización de dos páginas a la vez (como un libro o una revista abiertos) y una vuelta a la página animada.

- El segundo problema se encontró a la hora de compartir la ubicación actual. Al asociar al botón la acción "Compartir", hay que usar la palabra reservada "{location}" para obtener la ubicación actual, sin embargo, se encontró que había un bug para esta casuística en Mobincube y soporte tuvo que abrir una incidencia para resolver esto. Lo que estaba pasando es que se enviaba directamente el texto "{location}" en lugar de obtener las coordenadas. El bug estaba al intentar compartir la

ubicación al hacer click sobre un botón, sin embargo si se enviaba al hacer click sobre una imagen, el funcionamiento era correcto.

Al desconocer el tiempo de resolución de la incidencia, se tuvo que hacer un workaround para que la aplicación pudiera seguir contando con la funcionalidad de compartir la ubicación actual. Se creó una pantalla alternativa a la que se accediera al pulsar el botón “Mi Ubicación” del menú principal.



Ilustración 4.11: Workaround de envío de ubicación

Desde esta pantalla, al pulsar sobre la imagen con el texto “Enviar mi ubicación actual”, sí se podrían obtener las coordenadas de la ubicación dado que ya no se trata de hacer click sobre un botón, sino hacer click sobre una imagen.

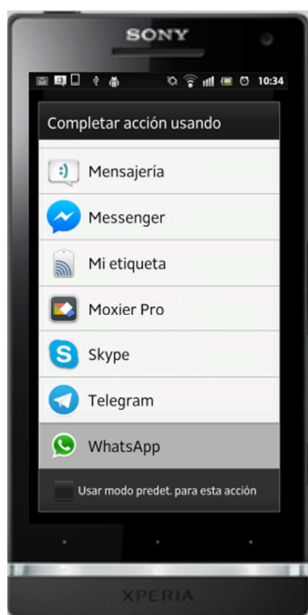


Ilustración 4.12.1: Envío de ubicación con workaround - Paso 1



Ilustración 4.12.2: Envío de ubicación con workaround - Paso 2



Ilustración 4.12.3: Envío de ubicación con workaround - Paso 3

Finalmente, antes de llevarse a cabo la entrega del presente proyecto final de carrera, se dio por resuelto el bug reportado a Mobincube por lo que se pudo comprobar que el comportamiento deseado ya funcionaba correctamente. Se duplicó la aplicación para añadir el comportamiento deseado suprimiendo la pantalla creada para el workaround y de este modo poder tener las aplicaciones con ambas alternativas.

➤ Pantalla Principal

Al abrir la aplicación se muestra una pantalla de publicidad de Mobincube.



Ilustración 4.13: Pantalla de inicio con publicidad de Mobincube

➤ Banner de publicidad en todas las pantallas



Ilustración 4.14: Banner de publicidad con Mobincube

➤ Manuales Emergencias

Al hacer click sobre alguno de ellos, se abrirá el Pdf accediendo a la url de issuu donde se encuentra almacenado. En este caso la visualización del documento será más lenta dado que dependerá de la velocidad de conexión a internet del dispositivo.

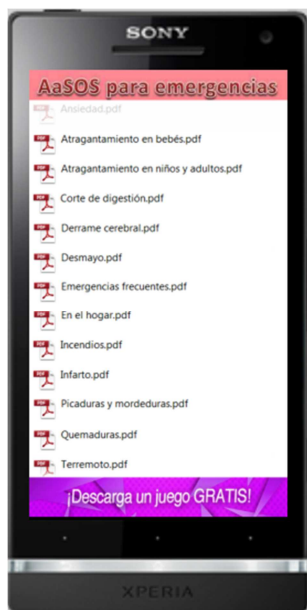


Ilustración 4.15.1:
Visualización de manuales con
Mobincube – Paso 1



Ilustración 4.15.2:
Visualización de manuales con
Mobincube – Paso 2



Ilustración 4.15.3:
Visualización de manuales con
Mobincube – Paso 3

4.3 Tutorial de desarrollo de AaSOS con MIT App Inventor 2

En este apartado se va a realizar un breve tutorial explicando paso a paso cómo se ha realizado el desarrollo de la pantalla principal de “Menú” con la lista de opciones y la opción de realizar llamada al número de emergencias 112. No obstante, en el Anexo I se pueden ver todos los bloques generados para desarrollar la aplicación completa.

- Paso 1: Crear nuevo proyecto.

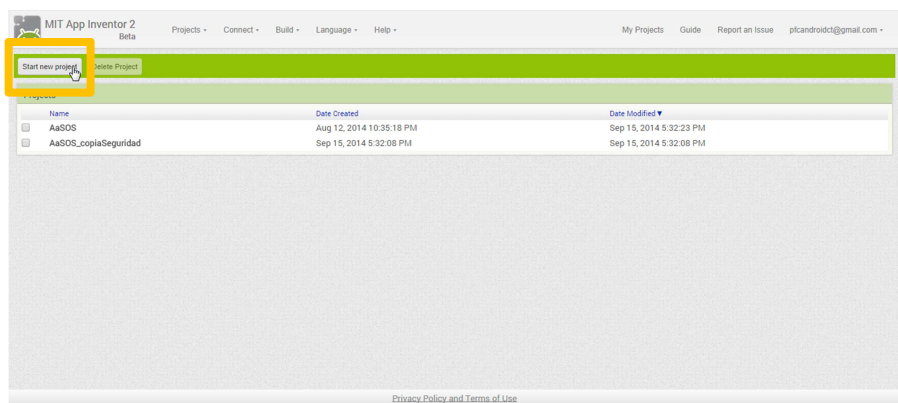


Ilustración 4.16: Tutorial MIT App Inventor 2 – Crear nuevo proyecto

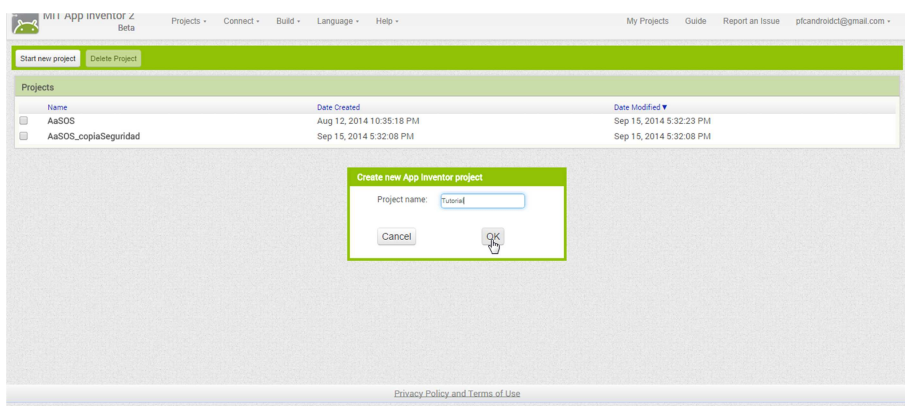


Ilustración 4.17: Tutorial MIT App Inventor 2 – Asignar nombre nuevo proyecto

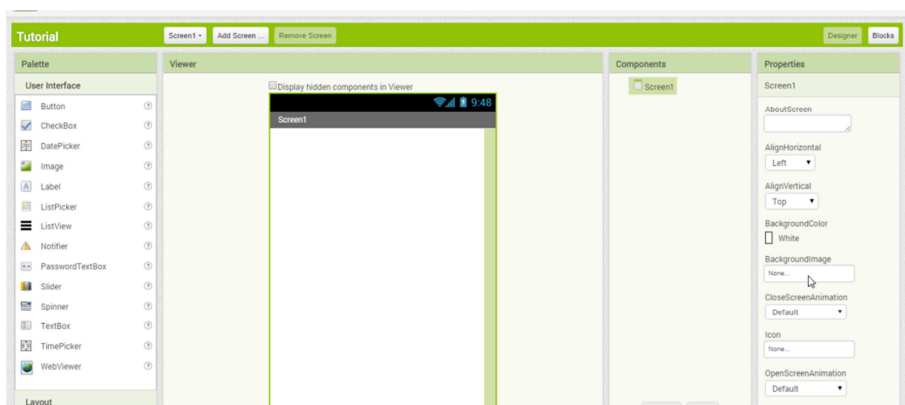


Ilustración 4.18: Tutorial MIT App Inventor 2 – Pantalla principal

- Paso 2: Añadir icono para la aplicación.

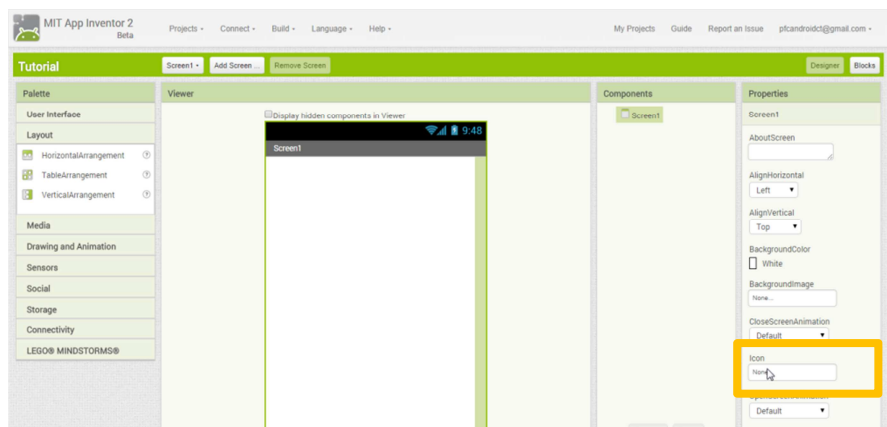


Ilustración 4.19: Tutorial MIT App Inventor 2 – Editar icono, paso 1

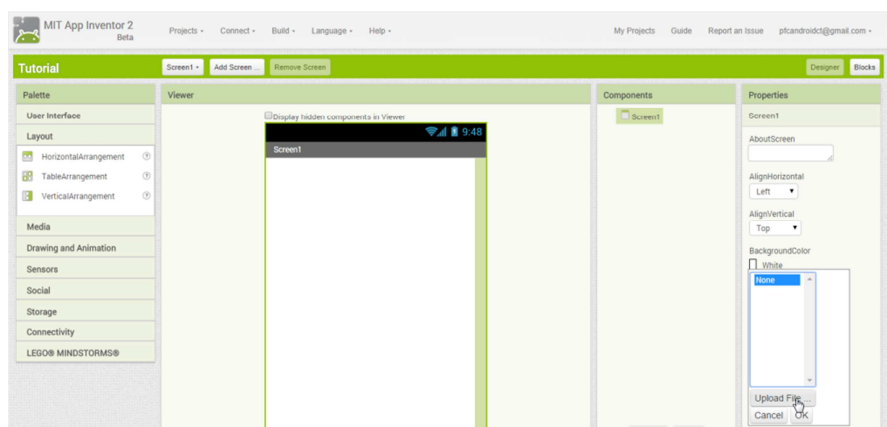


Ilustración 4.20: Tutorial MIT App Inventor 2 – Editar icono, paso 2

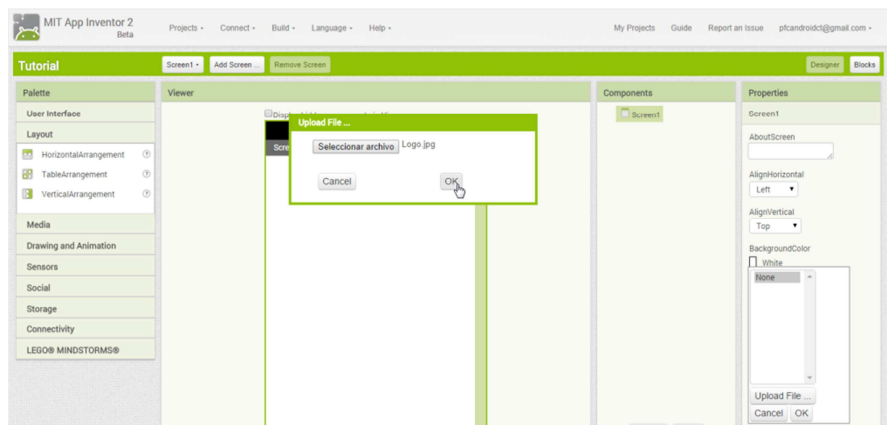


Ilustración 4.21: Tutorial MIT App Inventor 2 – Editar icono, paso 3

- Paso 3: Crear nueva pantalla para el Menú.

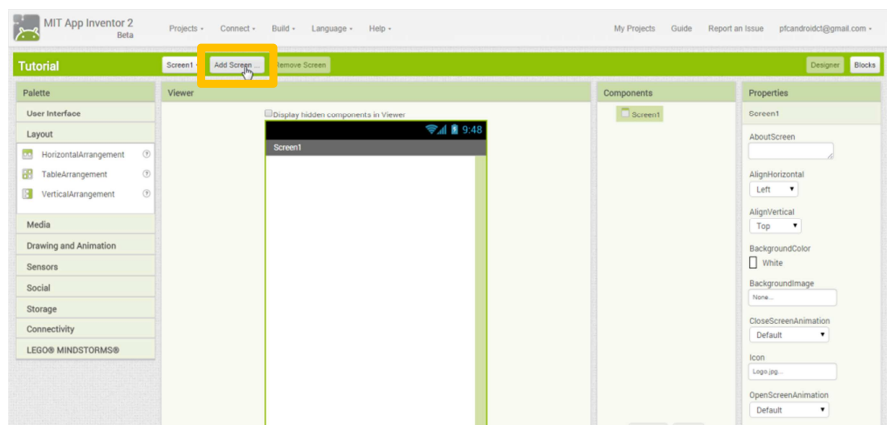


Ilustración 4.22: Tutorial MIT App Inventor 2 – Crear pantalla Menú, paso 1

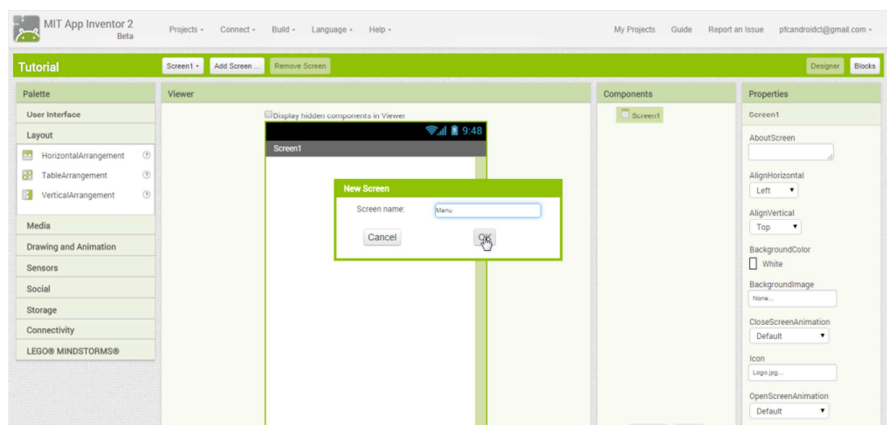


Ilustración 4.23: Tutorial MIT App Inventor 2 – Crear pantalla Menú, paso 2

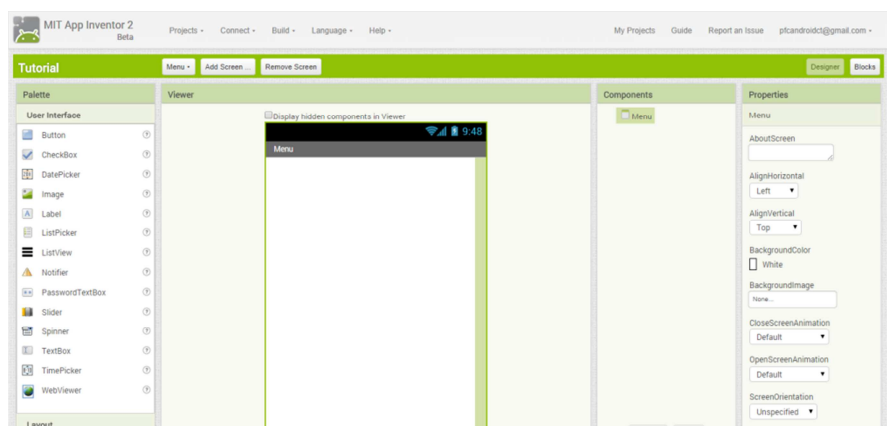


Ilustración 4.24: Tutorial MIT App Inventor 2 – Crear pantalla Menú, paso 3

- Paso 4: Elegir imagen de fondo.

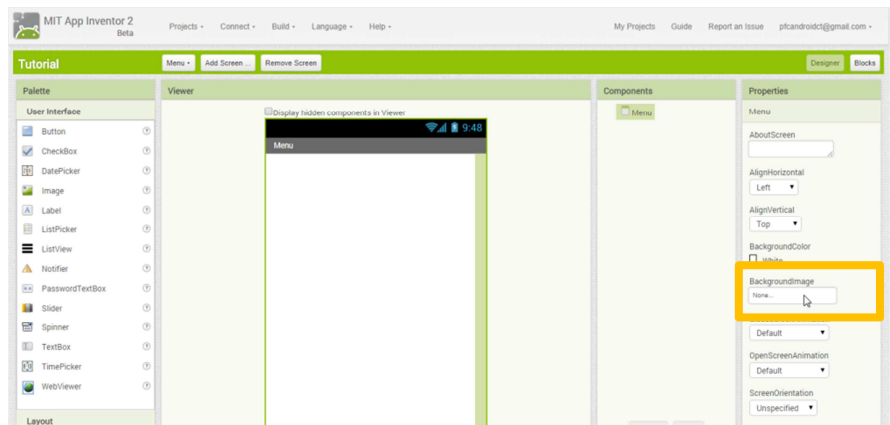


Ilustración 4.25: Tutorial MIT App Inventor 2 – Selección de imagen de fondo, paso 1

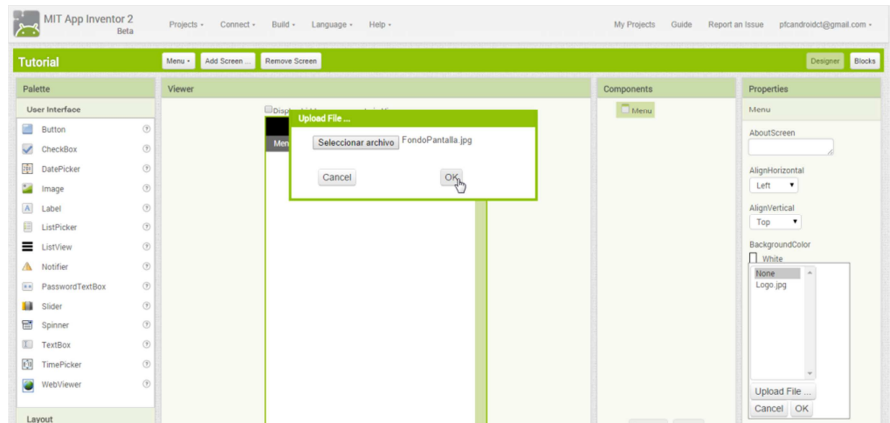


Ilustración 4.26: Tutorial MIT App Inventor 2 – Selección de imagen de fondo, paso 2

- Paso 5: Añadir cabecera arrastrando el objeto de tipo Image desde la paleta. Seleccionamos el objeto Image, arrastramos hasta la pantalla y soltamos. A continuación ajustamos el tamaño de la imagen.

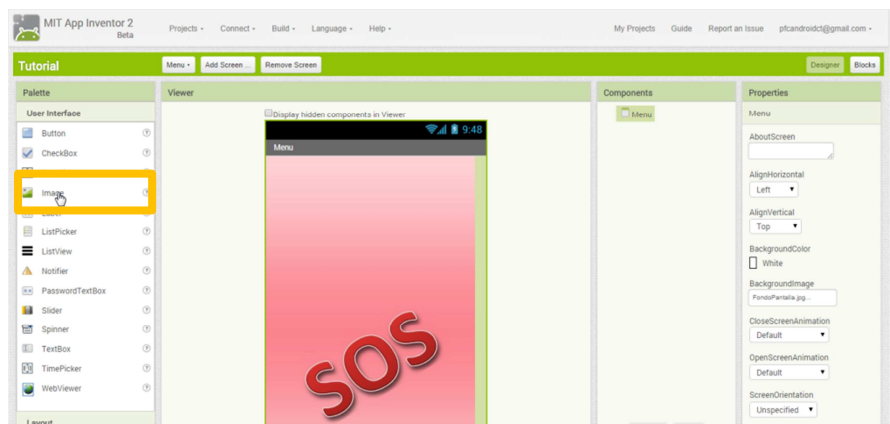


Ilustración 4.27: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 1

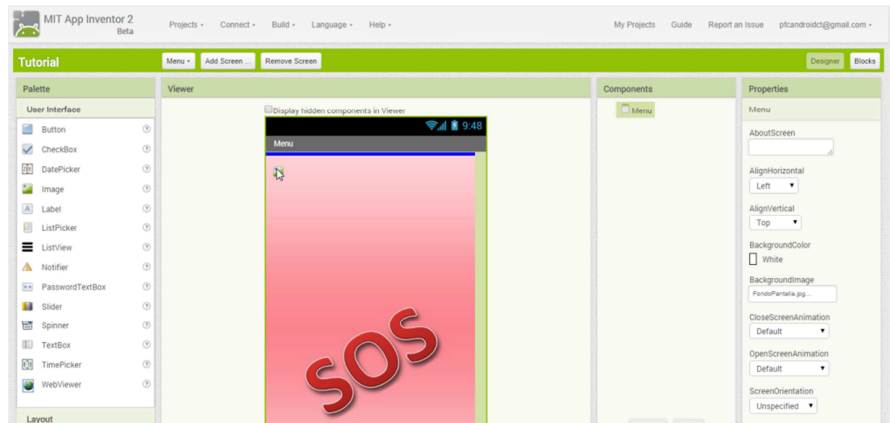


Ilustración 4.28: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 2

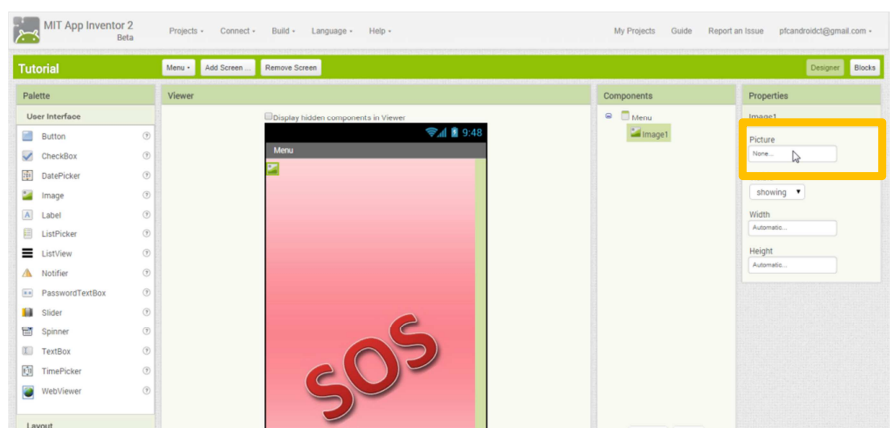


Ilustración 4.29: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 3

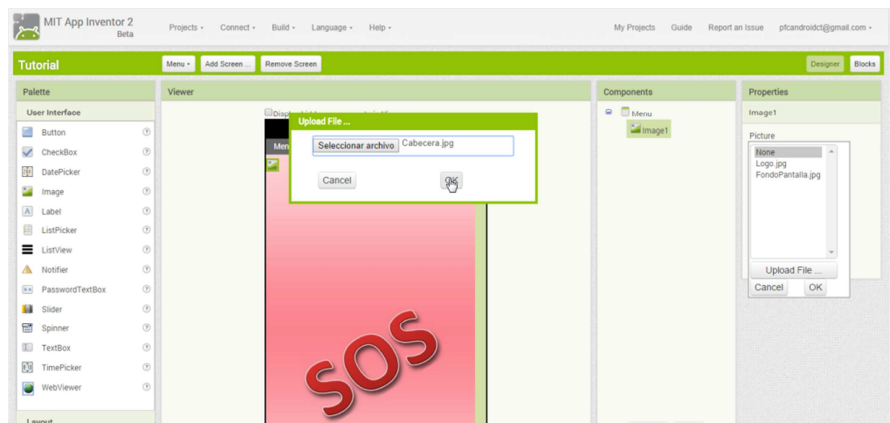


Ilustración 4.30: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 4

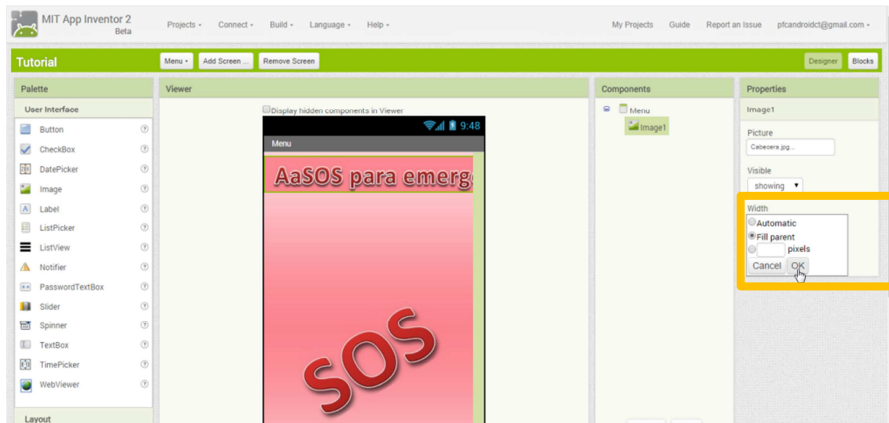


Ilustración 4.31: Tutorial MIT App Inventor 2 – Selección de imagen de cabecera, paso 5

- Paso 6: Añadir Layout horizontal.
Seleccionamos de la paleta el Layout, arrastramos y soltamos en el lugar deseado de la pantalla. Después ajustamos las propiedades con los valores deseados de anchura y altura.

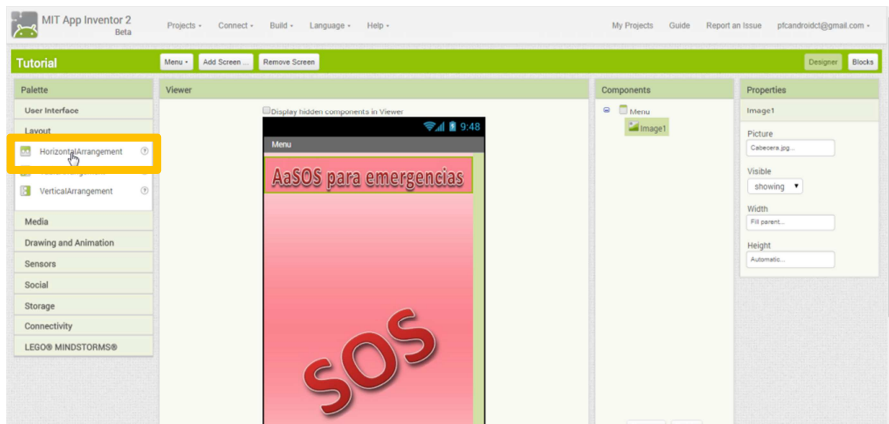


Ilustración 4.32: Tutorial MIT App Inventor 2 – Añadir Layout horizontal, paso 1

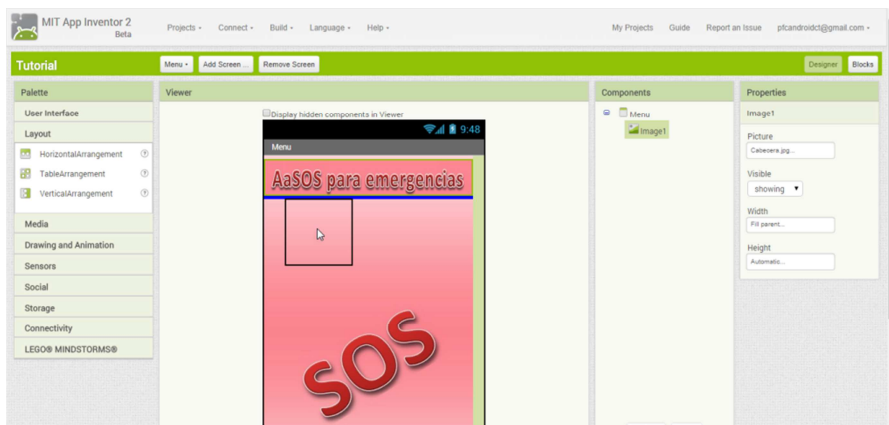


Ilustración 4.33: Tutorial MIT App Inventor 2 – Añadir Layout horizontal, paso 2

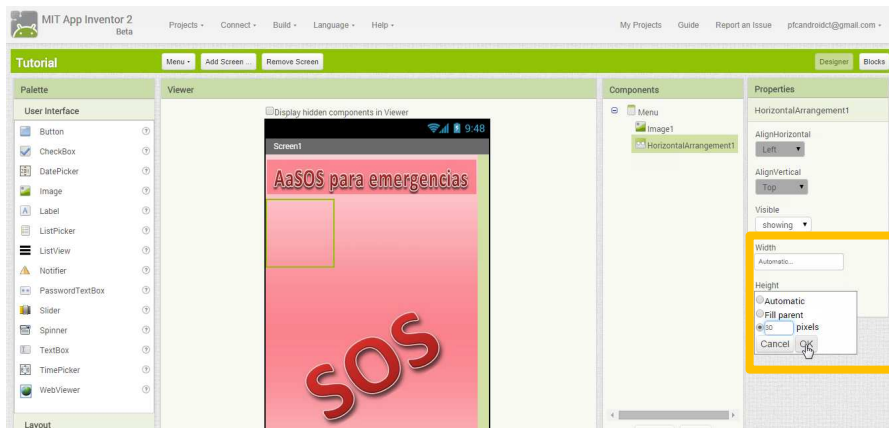


Ilustración 4.34: Tutorial MIT App Inventor 2 – Añadir Layout horizontal, paso 3

- Paso 7: Añadir opción de menú “Emergencias 112”.
Seleccionamos el objeto Button de la paleta y arrastramos y soltamos sobre la pantalla.



Ilustración 4.35: Tutorial MIT App Inventor 2 – Añadir opción de menú “Emergencias 112”, paso 1

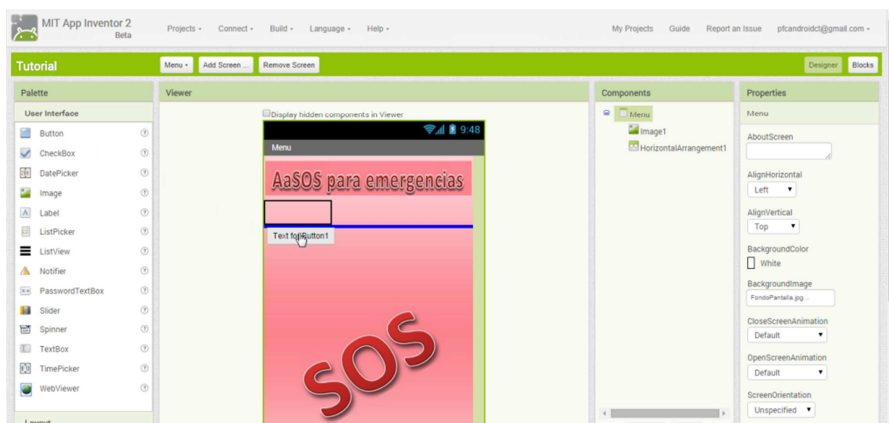


Ilustración 4.36: Tutorial MIT App Inventor 2 – Añadir opción de menú “Emergencias 112”, paso 2

- Paso 8: Ajustar propiedades del botón con los valores deseados.

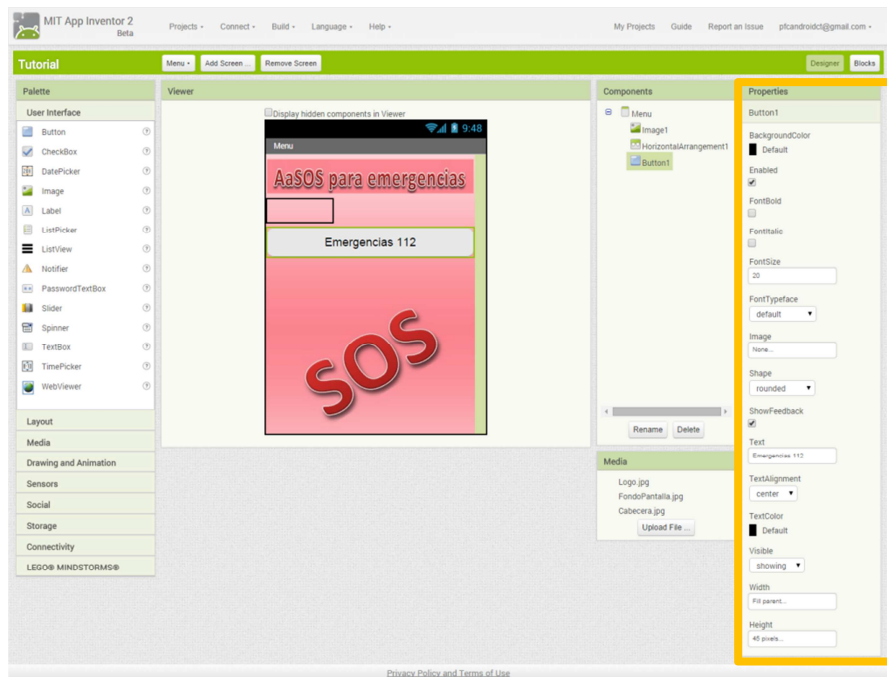


Ilustración 4.37: Tutorial MIT App Inventor 2 – Ajustar propiedades de botón “Emergencias 112”

- Paso 9: Añadir Layout y botón “Guardia Civil”.

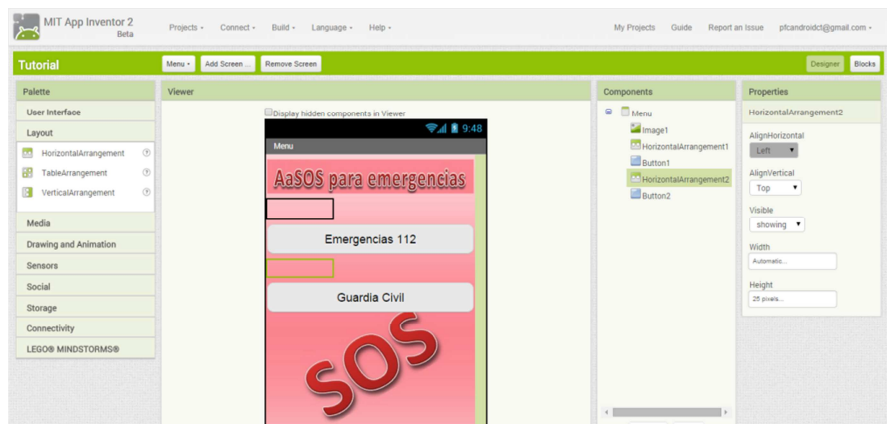


Ilustración 4.38: Tutorial MIT App Inventor 2 – Añadir Layout y botón “Guardia Civil”

- Paso 10: Añadir resto de opciones de menú del mismo modo que los anteriores.

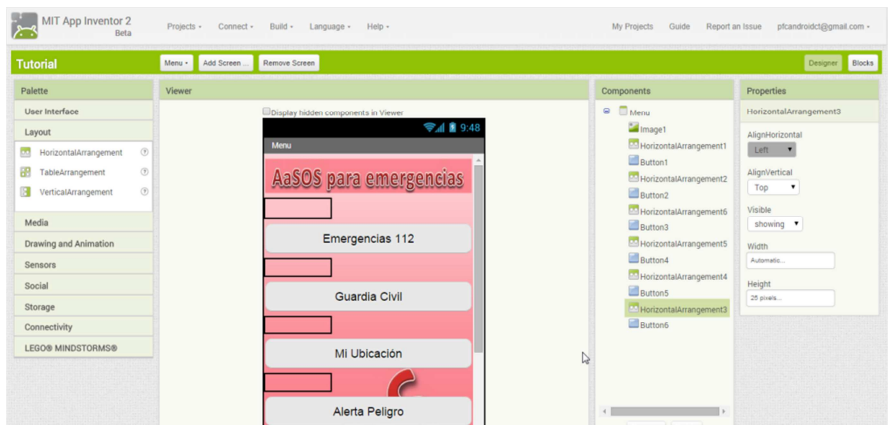


Ilustración 4.39: Tutorial MIT App Inventor 2 – Pantalla Menú completa

- Paso 11: Añadir desde la paleta el objeto *LocationSensor*, que será útil a la hora de asignar el comportamiento a la opción de menú “Mi Ubicación”.

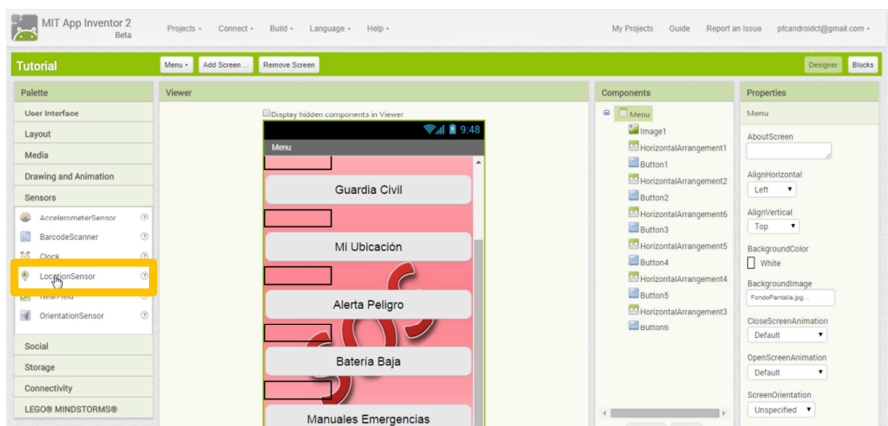


Ilustración 4.40: Tutorial MIT App Inventor 2 – Selección de objeto LocationSensor, paso 1

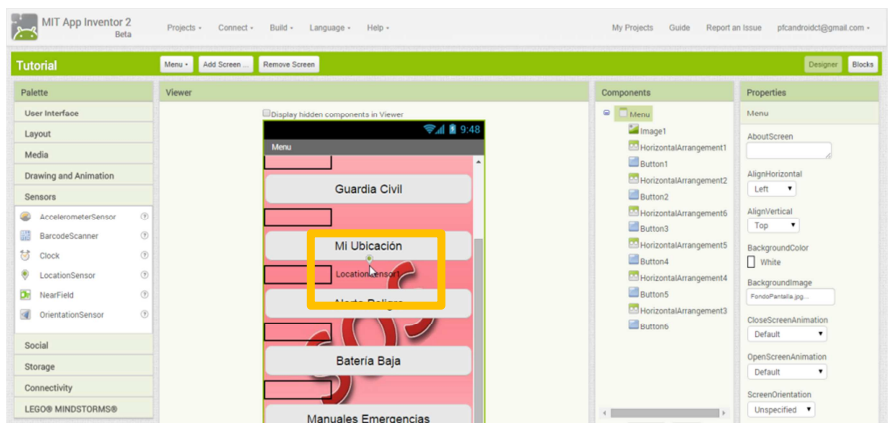


Ilustración 4.41: Tutorial MIT App Inventor 2 – Selección de objeto LocationSensor, paso 2

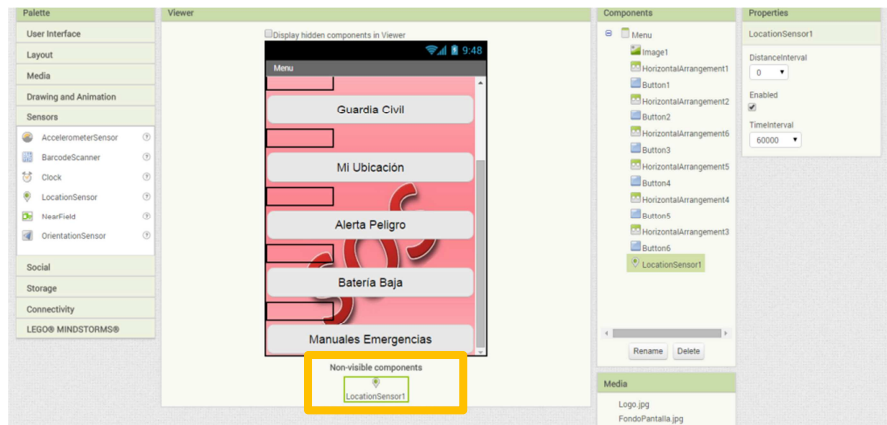


Ilustración 4.42: Tutorial MIT App Inventor 2 – Selección de objeto LocationSensor, paso 3

- Paso 12: Añadir desde la paleta el objeto *PhoneCall*, que será útil a la hora de asignar el comportamiento de las opciones de menú “Emergencias 112” y “Guardia Civil” desde las cuales se realizan llamadas telefónicas.

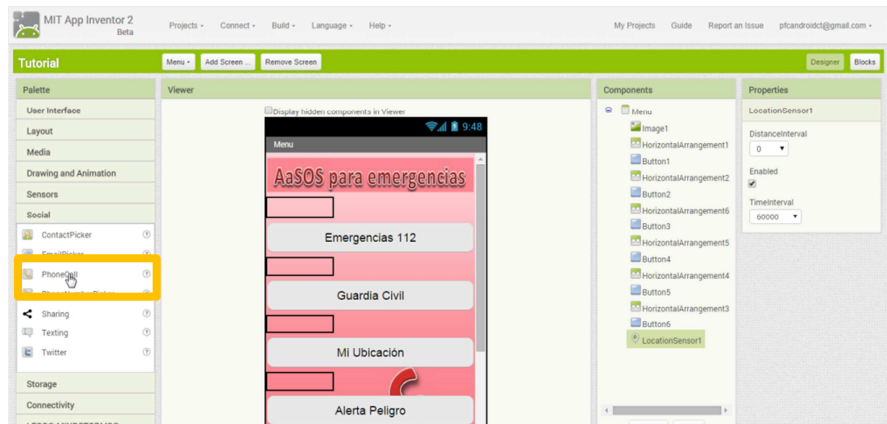


Ilustración 4.43: Tutorial MIT App Inventor 2 – Selección de objeto PhoneCall, paso 1

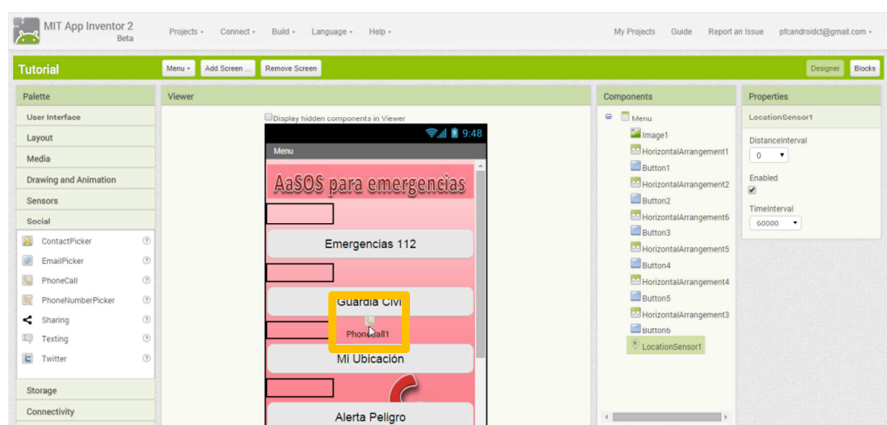


Ilustración 4.44: Tutorial MIT App Inventor 2 – Selección de objeto PhoneCall, paso 2

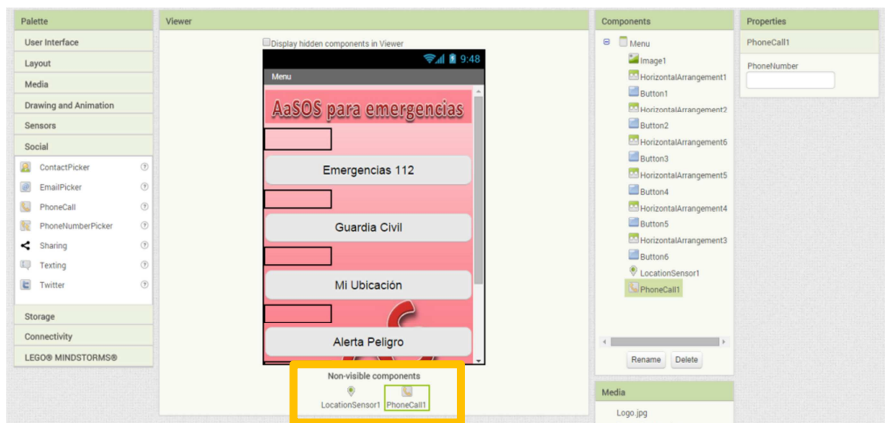


Ilustración 4.45: Tutorial MIT App Inventor 2 – Selección de objeto PhoneCall, paso 3

- Paso 13: Cambiar nombres de los botones para poder identificarlos fácilmente con la opción Rename.

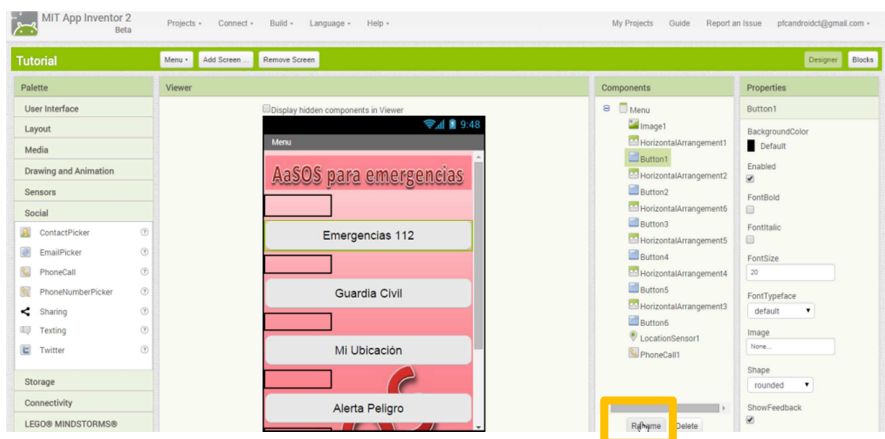


Ilustración 4.46: Tutorial MIT App Inventor 2 – Renombrar botones

- Paso 14: Ir a la vista de bloques para desarrollar el comportamiento de la aplicación.

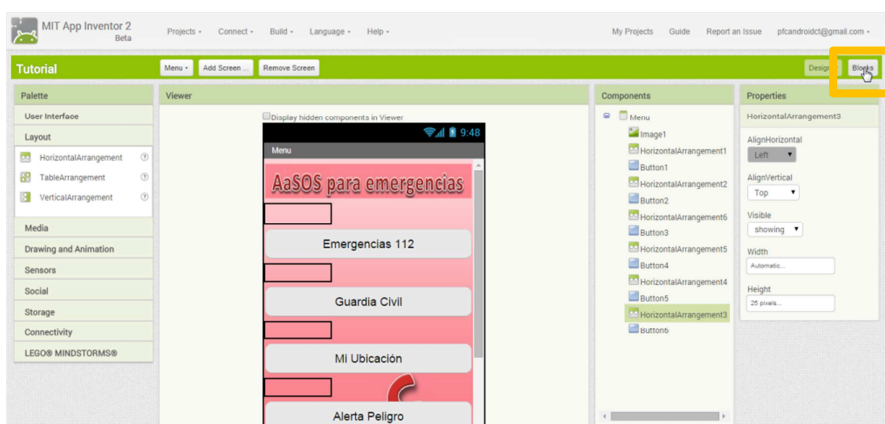


Ilustración 4.47: Tutorial MIT App Inventor 2 – Acceso a editor de bloques

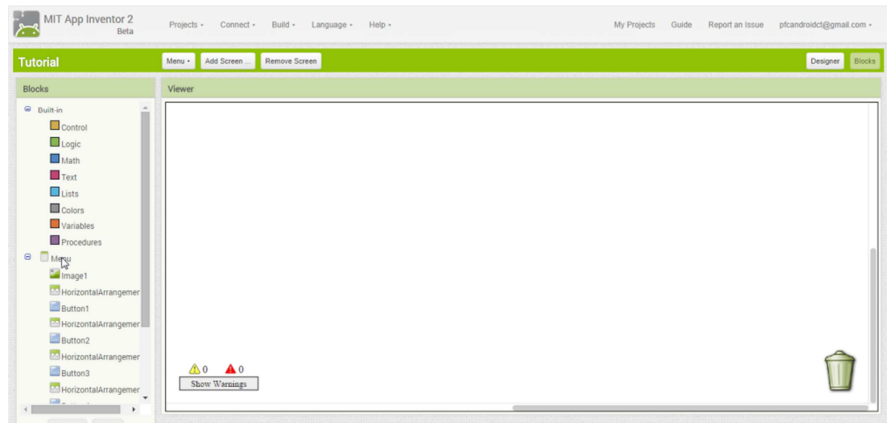


Ilustración 4.48: Tutorial MIT App Inventor 2 – Editor de bloques

- Paso 15: Añadir bloque para detectar evento cuando se inicialice la pantalla de Menú.
Se selecciona el evento deseado y se arrastra a la sección Viewer.

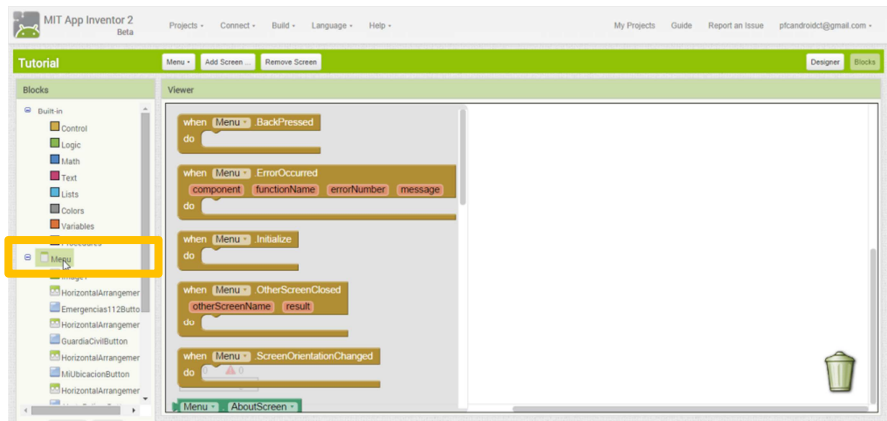


Ilustración 4.49: Tutorial MIT App Inventor 2 – Evento inicialización Menú, paso 1

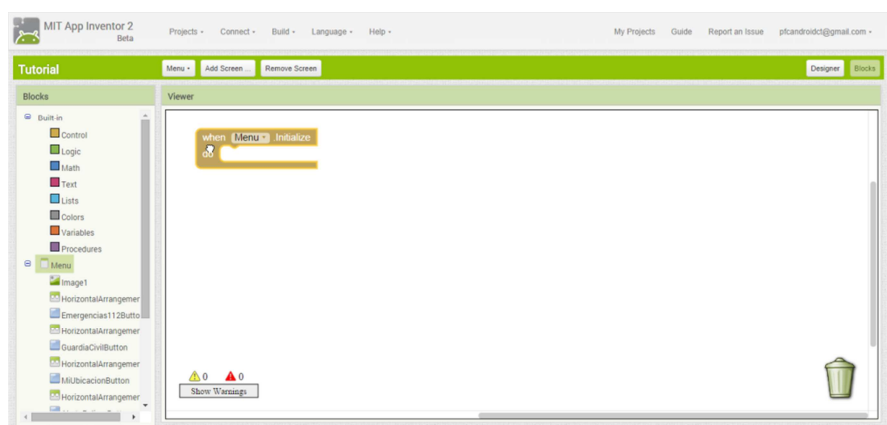


Ilustración 4.50: Tutorial MIT App Inventor 2 – Evento inicialización Menú, paso 2

- Paso 16: Añadir comportamiento para el evento seleccionado.
En primer lugar se deshabilita el sensor de localización para activarlo únicamente cuando se pulse la opción de menú “Mi Ubicación”. A continuación se cambia el color de fondo y letra de cada uno de los botones en la pantalla de Menú.

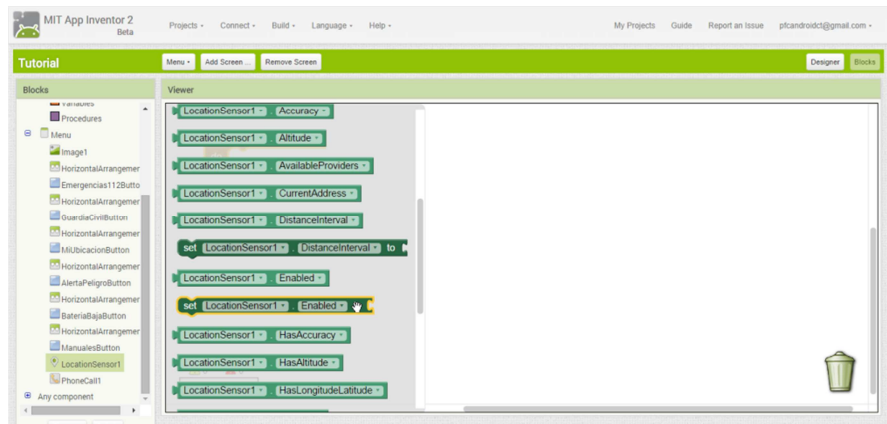


Ilustración 4.51: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 1

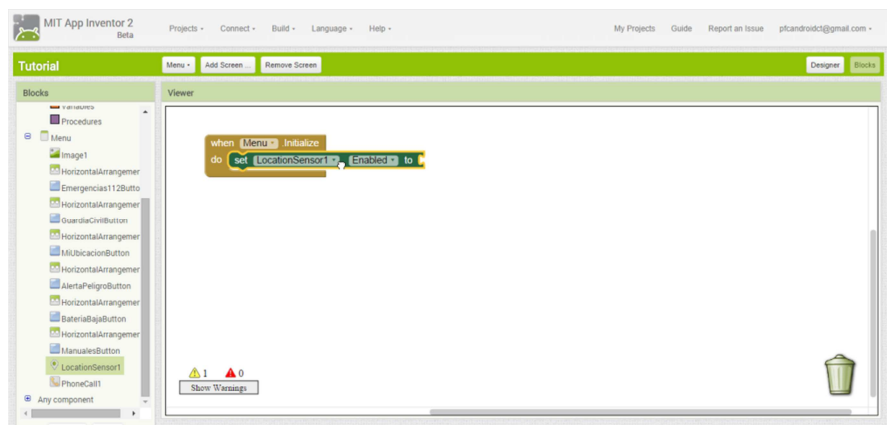


Ilustración 4.52: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 2

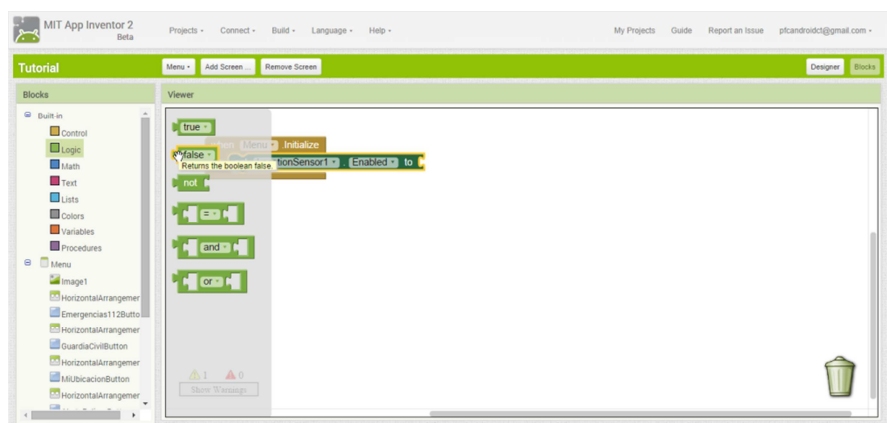


Ilustración 4.53: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 3

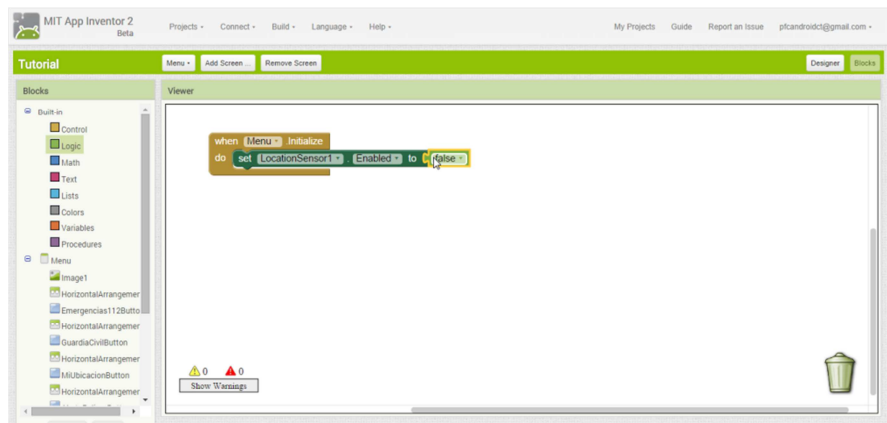


Ilustración 4.54: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 4

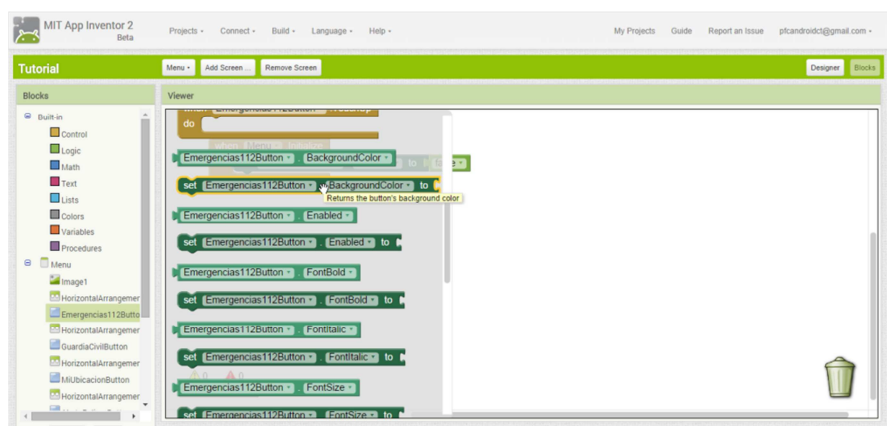


Ilustración 4.55: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 5

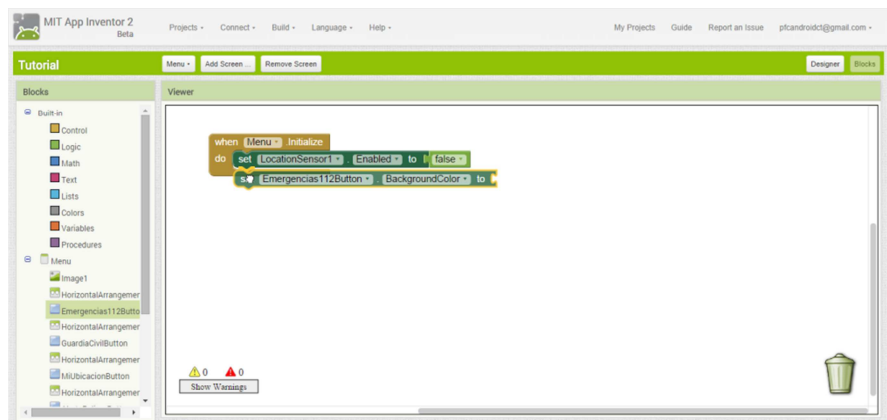


Ilustración 4.56: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 6

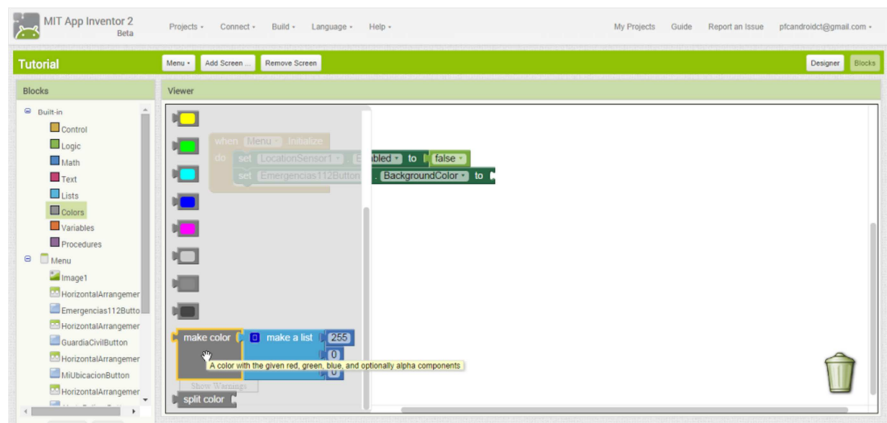


Ilustración 4.57: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 7

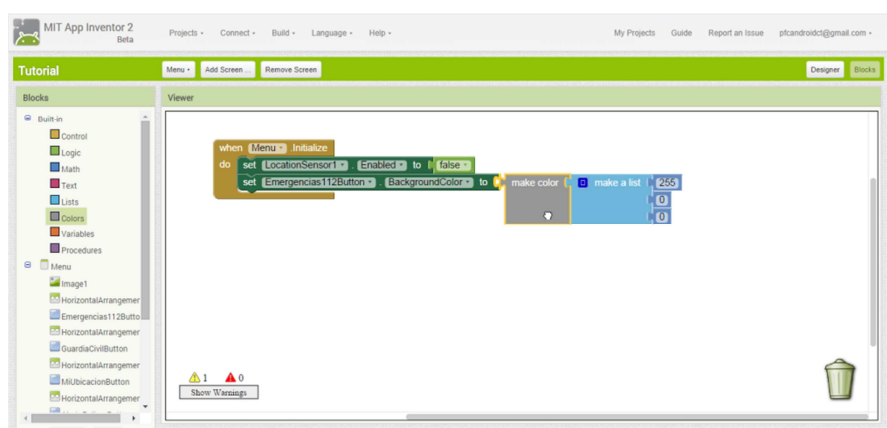


Ilustración 4.58: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 8

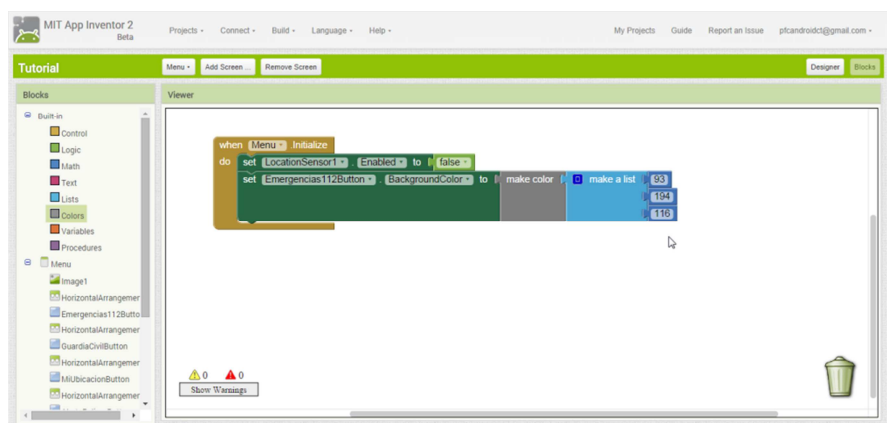
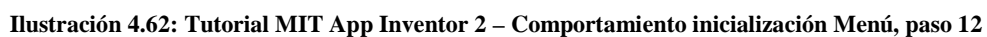
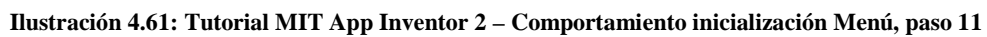
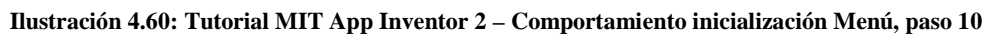


Ilustración 4.59: Tutorial MIT App Inventor 2 – Comportamiento inicialización Menú, paso 9



- Paso 17: Añadir comentario al comportamiento desarrollado.

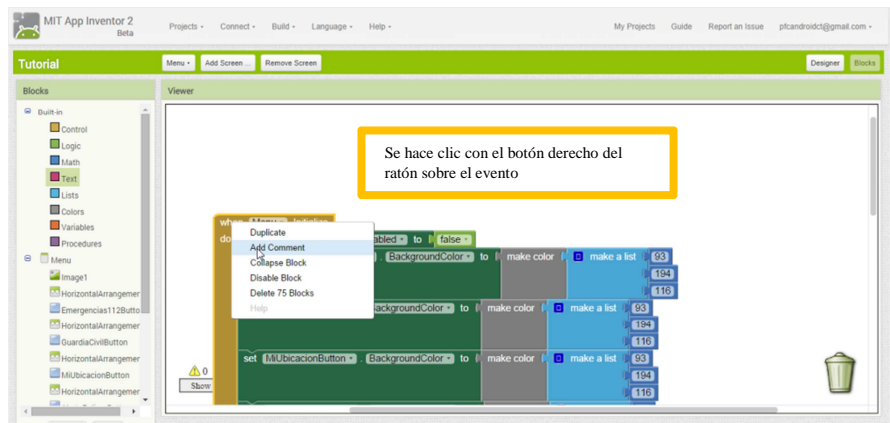


Ilustración 4.63: Tutorial MIT App Inventor 2 – Añadir comentario, paso 1

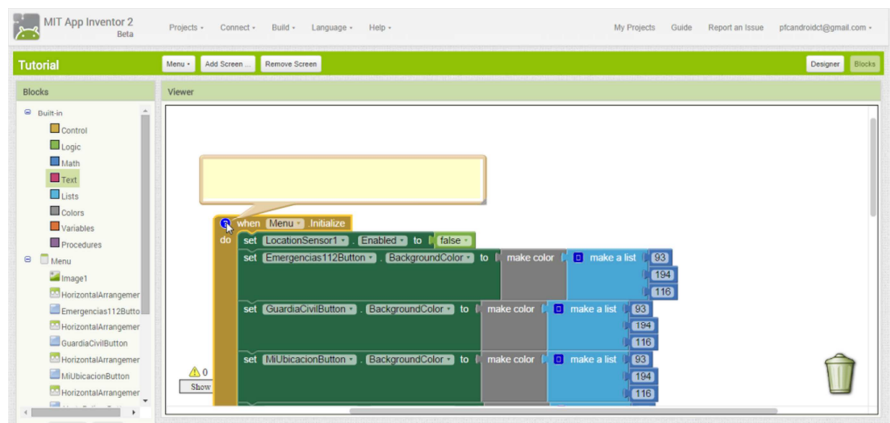


Ilustración 4.64: Tutorial MIT App Inventor 2 – Añadir comentario, paso 2

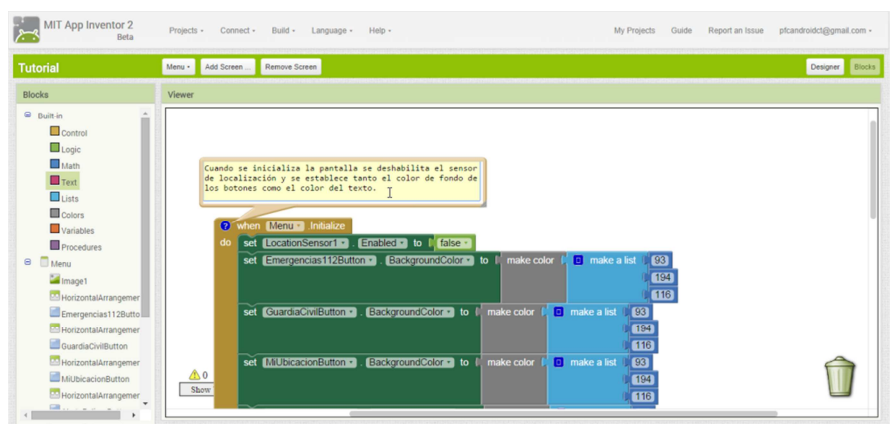


Ilustración 4.65: Tutorial MIT App Inventor 2 – Añadir comentario, paso 3

- Paso 18: Añadir bloque para detectar evento click de la opción de menú “Emergencias 112”.

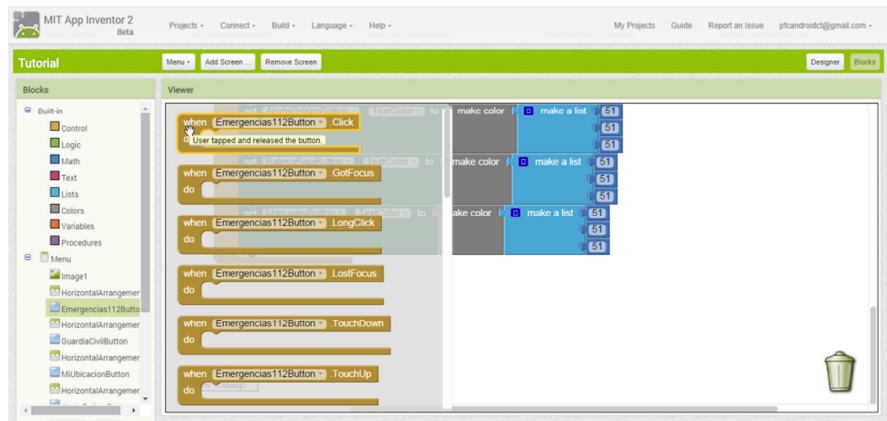


Ilustración 4.66: Tutorial MIT App Inventor 2 – Evento Click de botón “Emergencias112Button”, paso 1

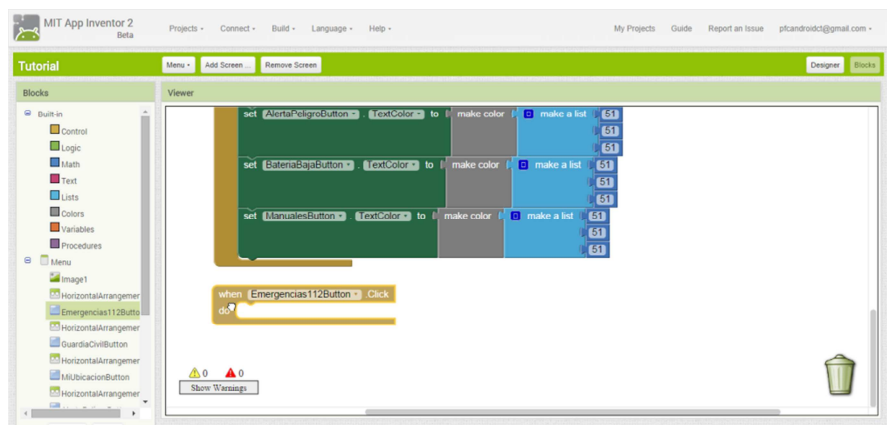


Ilustración 4.67: Tutorial MIT App Inventor 2 – Evento Click de botón “Emergencias112Button”, paso 2

- Paso 19: Añadir comportamiento al hacer click en la opción de menú “Emergencias 112”. En primer lugar se configura el número de teléfono y a continuación se añade el evento que inicia la llamada telefónica.

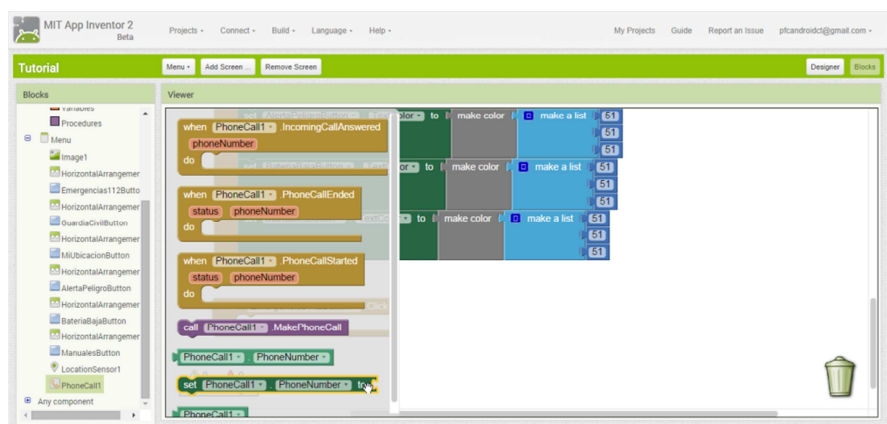


Ilustración 4.68: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 1

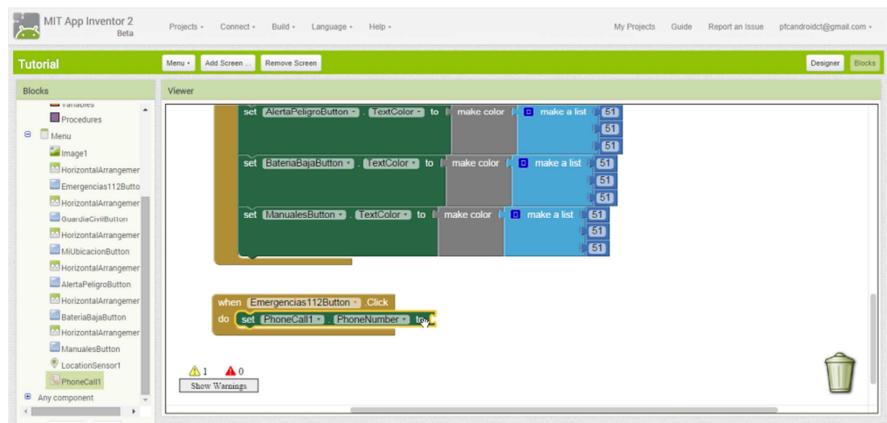


Ilustración 4.69: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 2

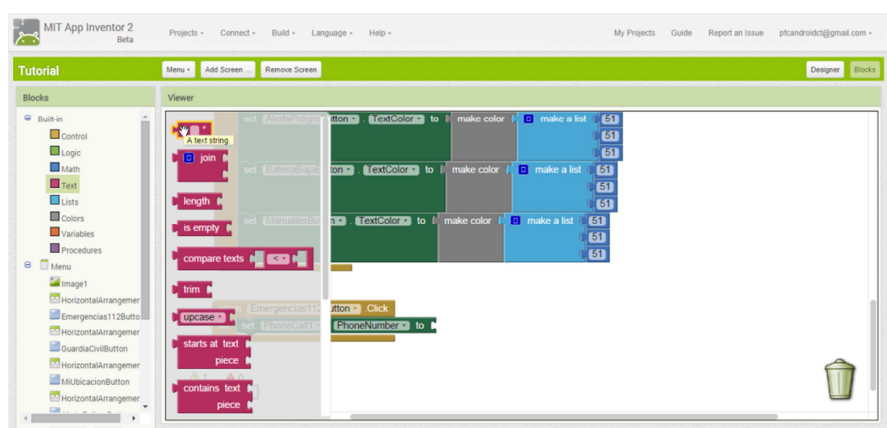


Ilustración 4.70: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 3



Ilustración 4.71: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 4

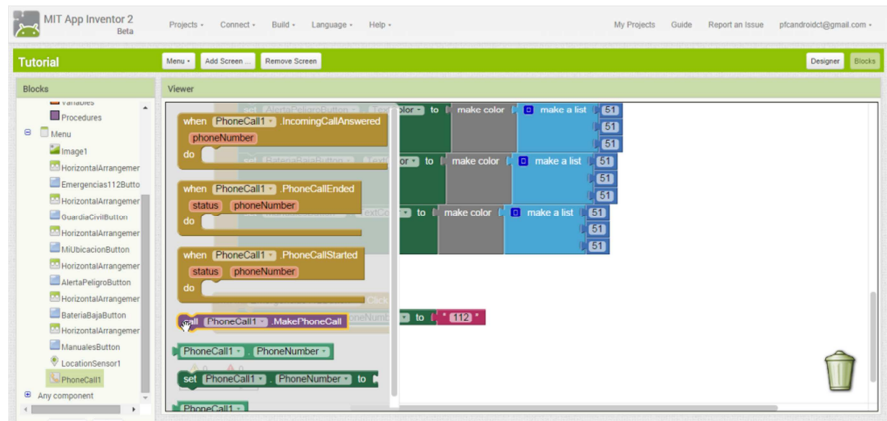


Ilustración 4.72: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 5

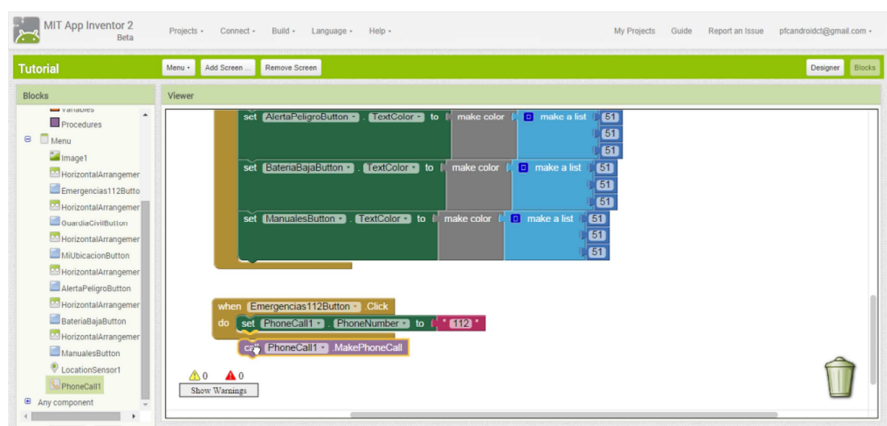


Ilustración 4.73: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 6

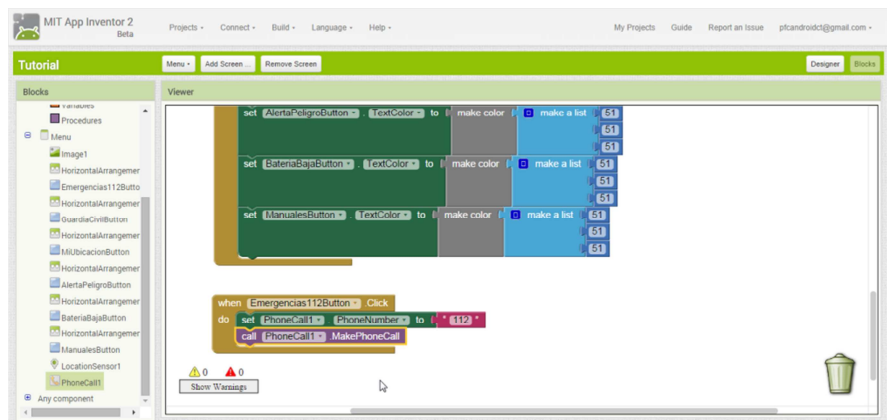


Ilustración 4.74: Tutorial MIT App Inventor 2 – Comportamiento botón “Emergencias 112”, paso 7

4.4 Tutorial de desarrollo de AaSOS con Mobincube

En este apartado se va a realizar un breve tutorial explicando paso a paso cómo se ha realizado el desarrollo de la pantalla principal de “Menú” con la lista de opciones y se detallará cómo asignar la acción de realizar llamada telefónica. No obstante, en el Anexo II se puede ver el detalle completo de cómo se ha desarrollado la aplicación.

- Paso 1: Crear nuevo proyecto.

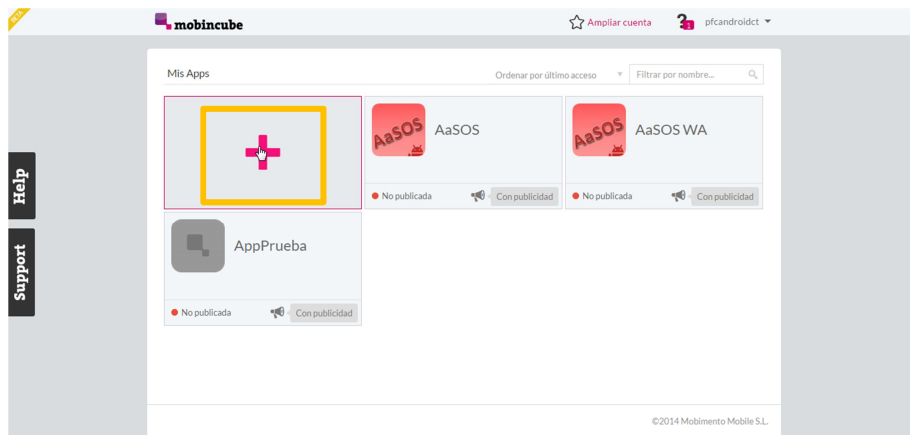


Ilustración 4.75: Tutorial Mobincube – Crear nuevo proyecto

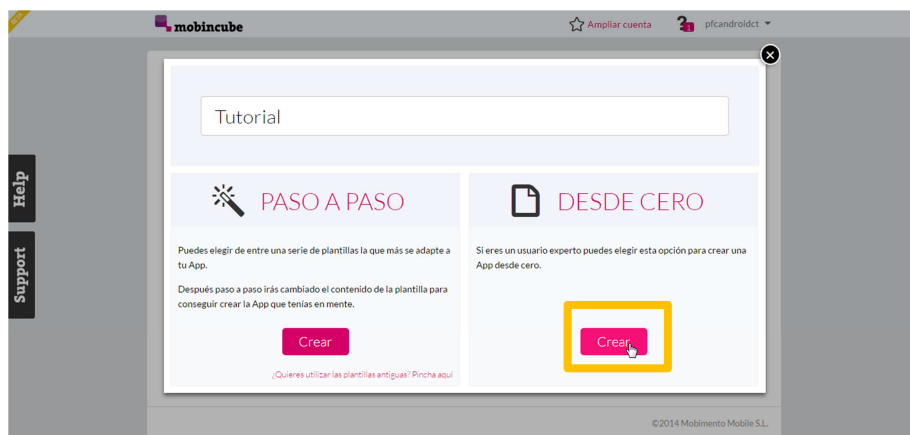


Ilustración 4.76: Tutorial Mobincube – Asignar nombre nuevo proyecto

- Paso 2: Añadir icono para la aplicación.

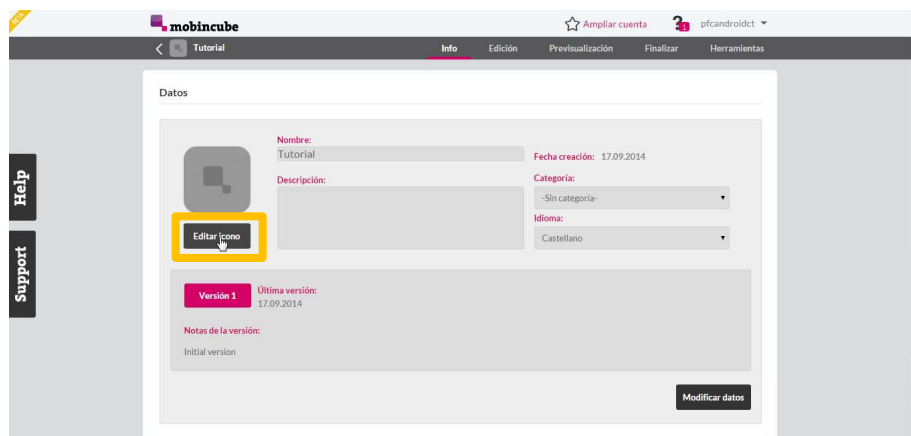


Ilustración 4.77: Tutorial Mobincube – Editar icono, paso 1

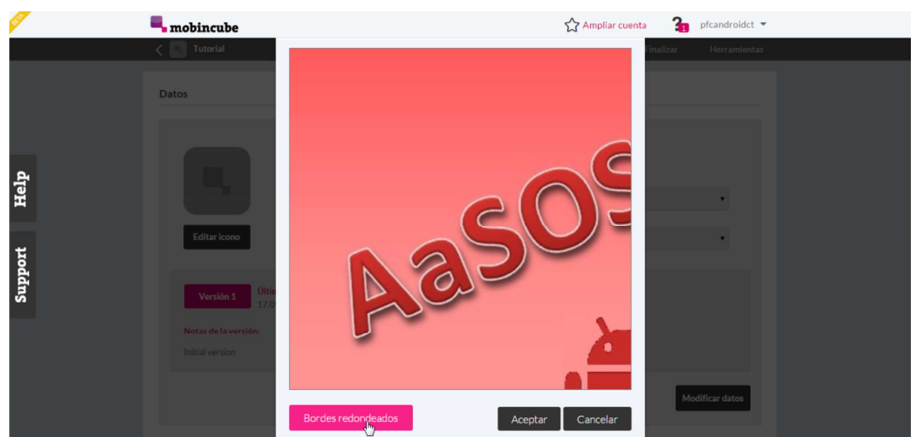


Ilustración 4.78: Tutorial Mobincube – Editar icono, paso 2

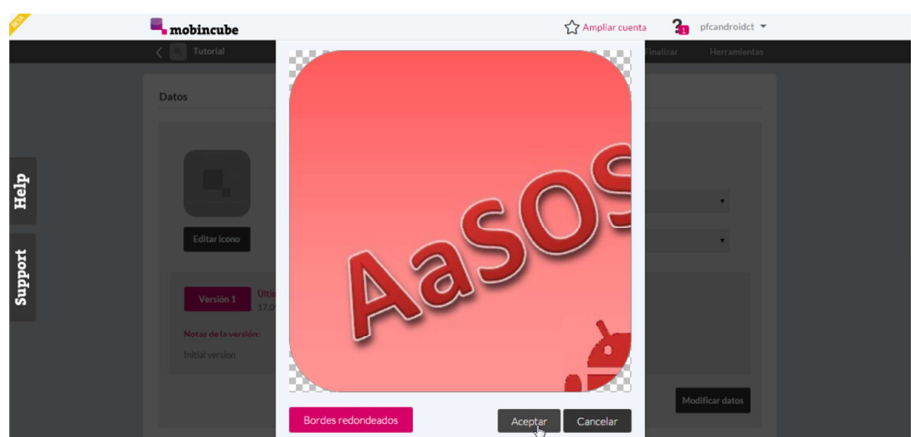


Ilustración 4.79: Tutorial Mobincube – Editar icono, paso 3

- Paso 3: Editar información de la aplicación

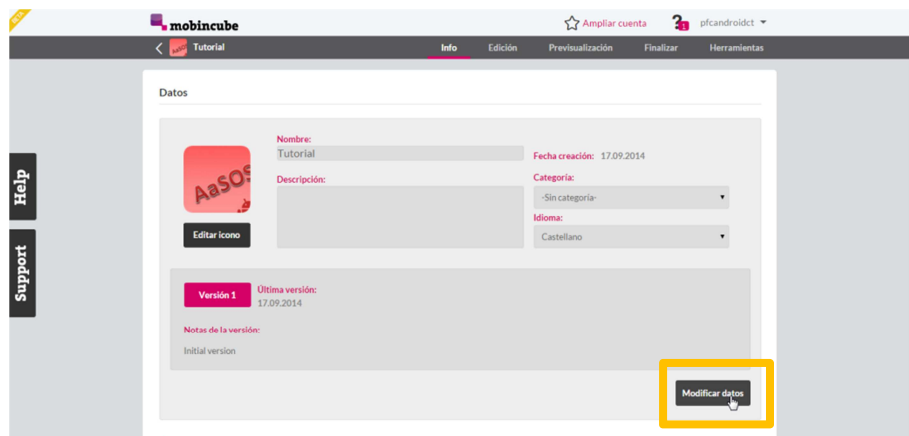


Ilustración 4.80: Tutorial Mobincube – Modificar datos app, paso 1

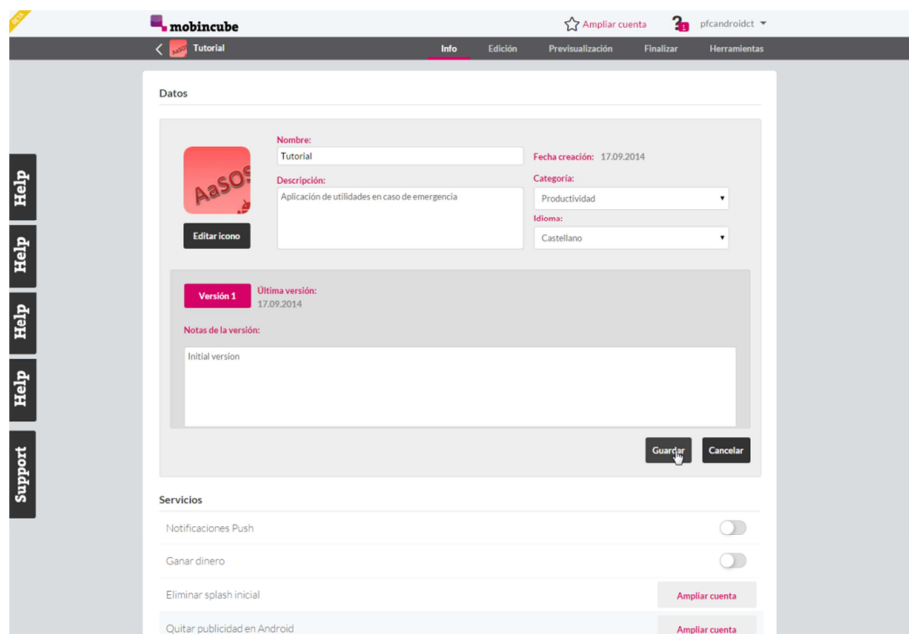


Ilustración 4.81: Tutorial Mobincube – Modificar datos app, paso 2

- Paso 4: Edición de la aplicación

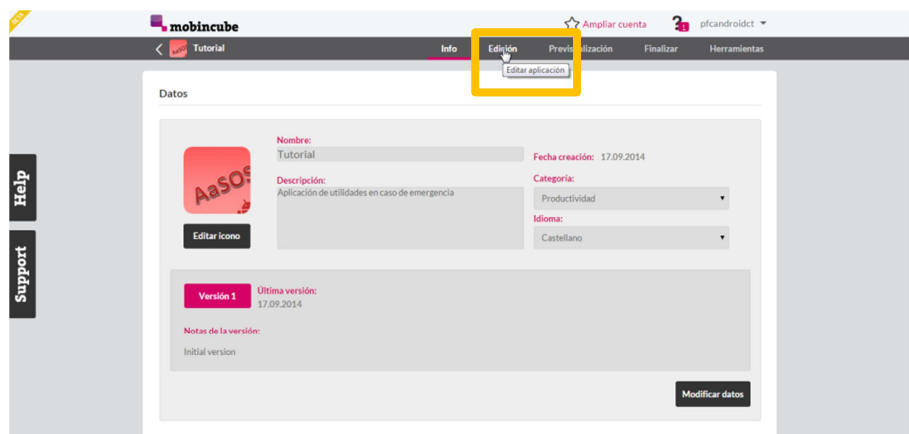


Ilustración 4.82: Tutorial Mobincube – Edición app

- Paso 5: Selección de pantalla inicial

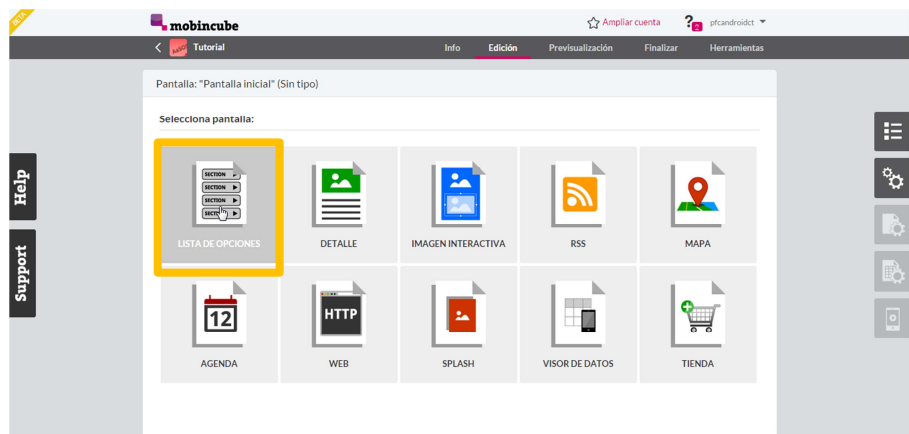


Ilustración 4.83: Tutorial Mobincube – Selección de pantalla inicial

- Paso 6: Elegir imagen de fondo.

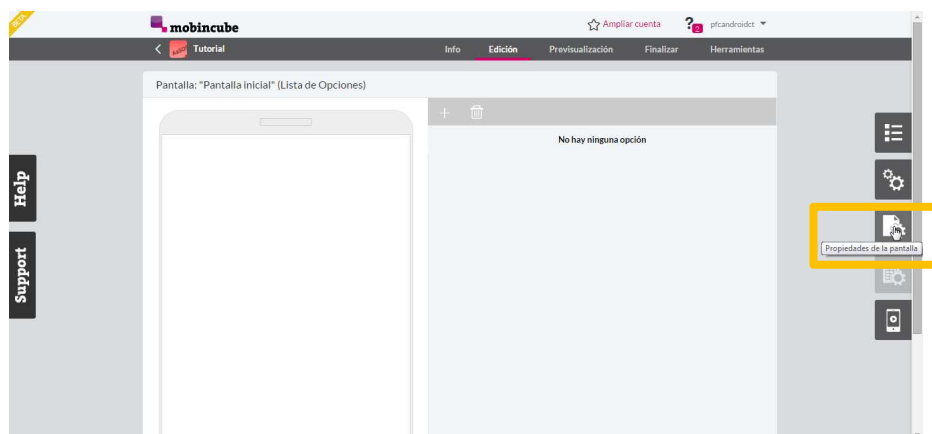


Ilustración 4.84: Tutorial Mobincube – Selección de imagen de fondo, paso 1

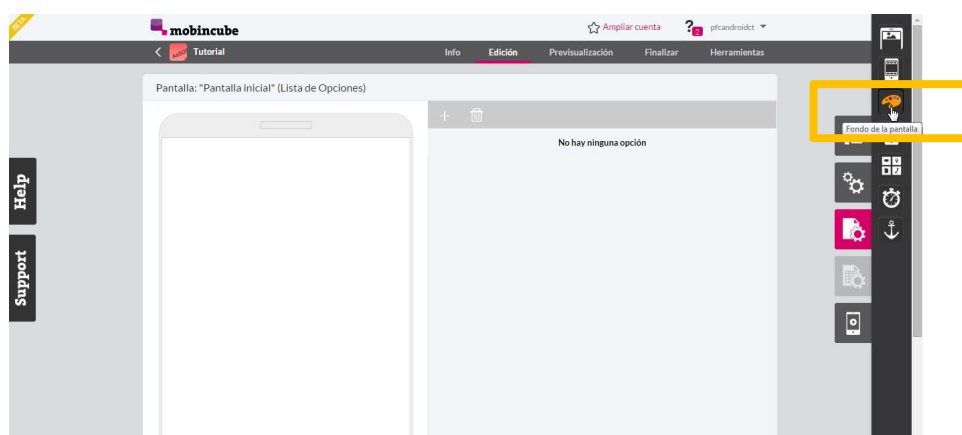


Ilustración 4.85: Tutorial Mobincube – Selección de imagen de fondo, paso 2

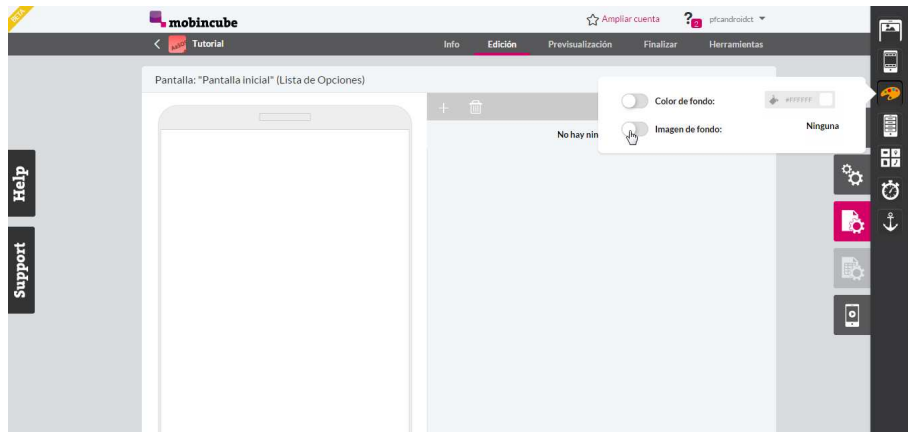


Ilustración 4.86: Tutorial Mobincube – Selección de imagen de fondo, paso 3

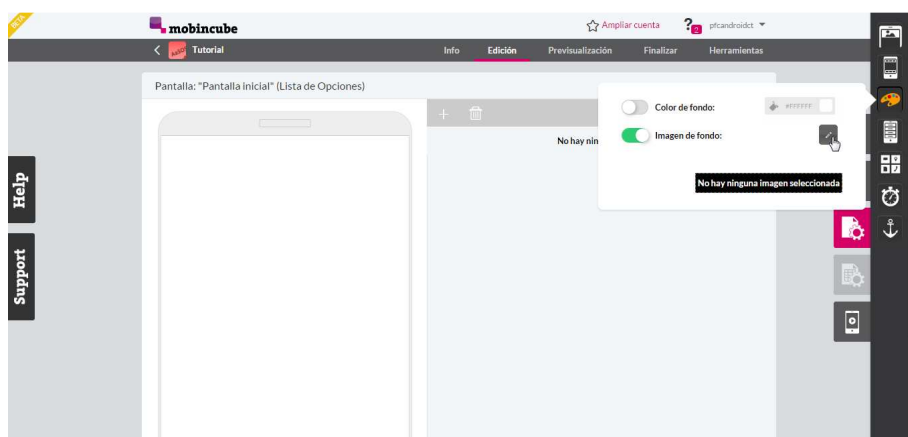


Ilustración 4.87: Tutorial Mobincube – Selección de imagen de fondo, paso 4

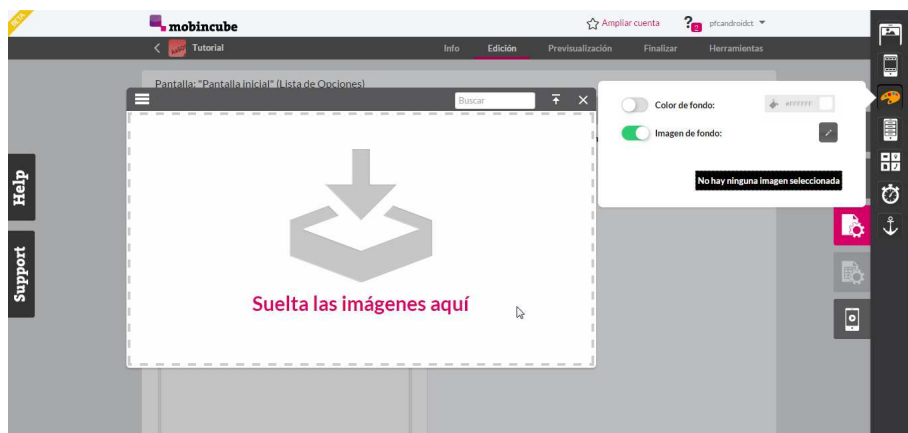


Ilustración 4.88: Tutorial Mobincube – Selección de imagen de fondo, paso 5

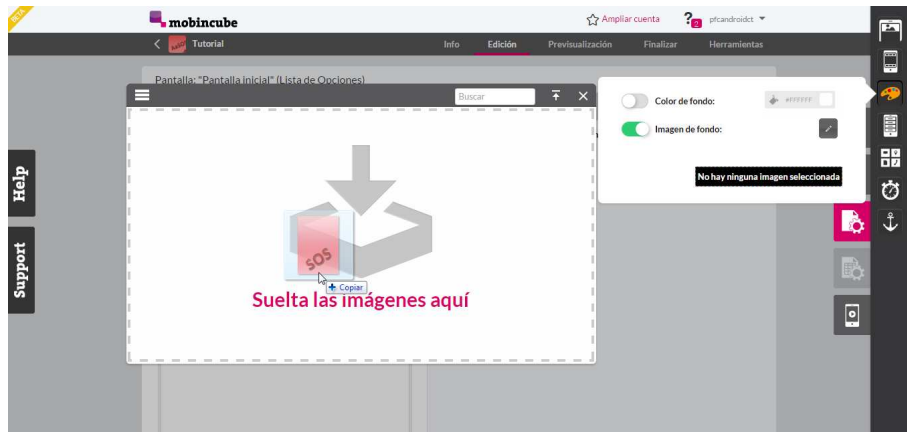


Ilustración 4.89: Tutorial Mobincube – Selección de imagen de fondo, paso 6

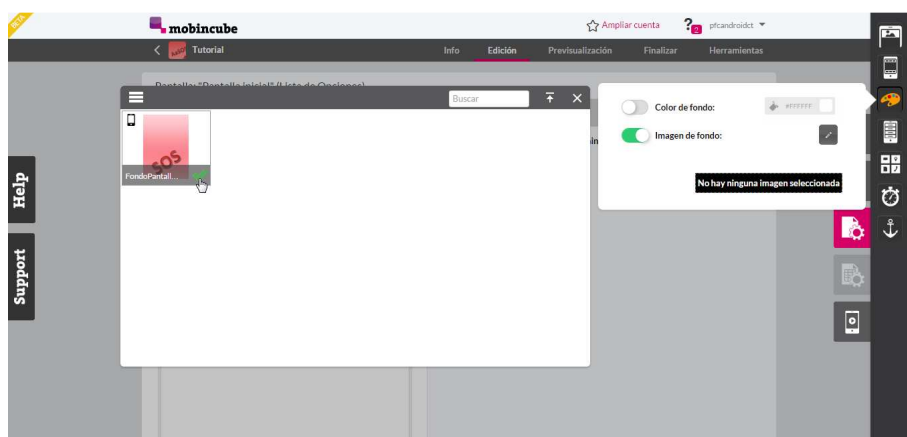


Ilustración 4.90: Tutorial Mobincube – Selección de imagen de fondo, paso 7

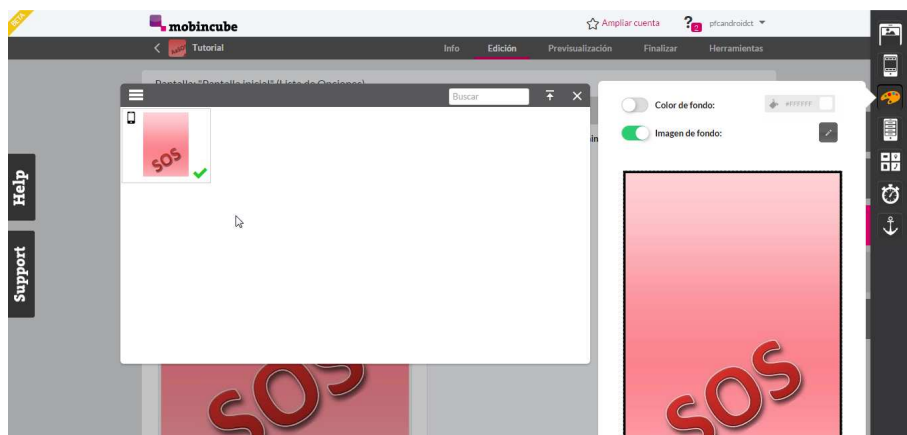


Ilustración 4.91: Tutorial Mobincube – Selección de imagen de fondo, paso 8

- Paso 7: Añadir cabecera.

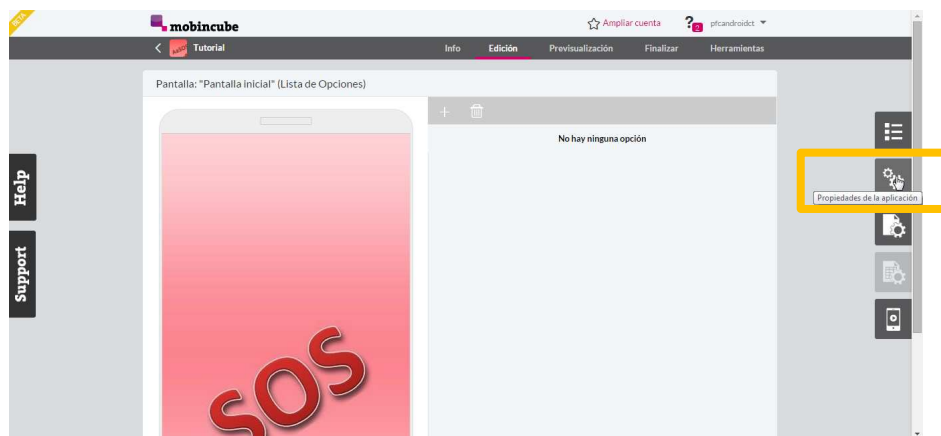


Ilustración 4.92: Tutorial Mobincube – Selección de imagen de cabecera, paso 1

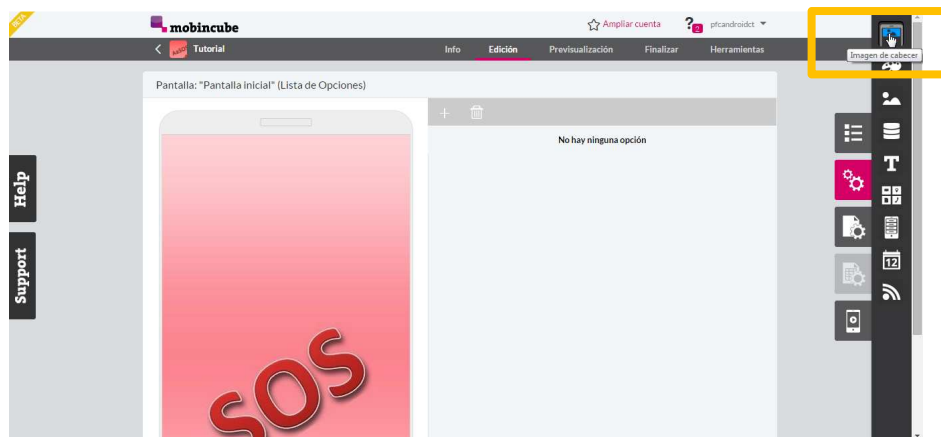


Ilustración 4.93: Tutorial Mobincube – Selección de imagen de cabecera, paso 2

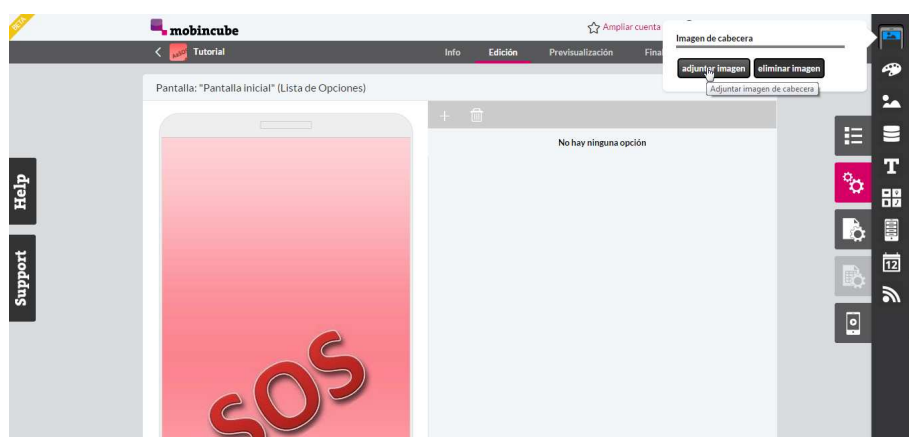


Ilustración 4.94: Tutorial Mobincube – Selección de imagen de cabecera, paso 3

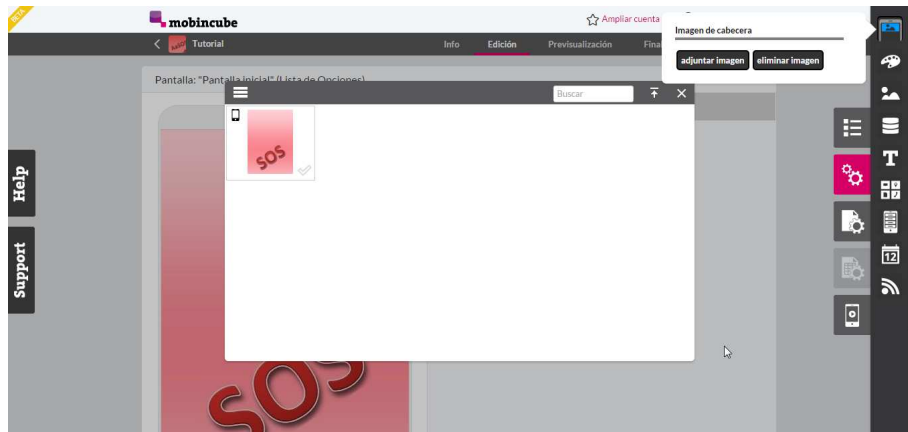


Ilustración 4.95: Tutorial Mobincube – Selección de imagen de cabecera, paso 4



Ilustración 4.96: Tutorial Mobincube – Selección de imagen de cabecera, paso 5

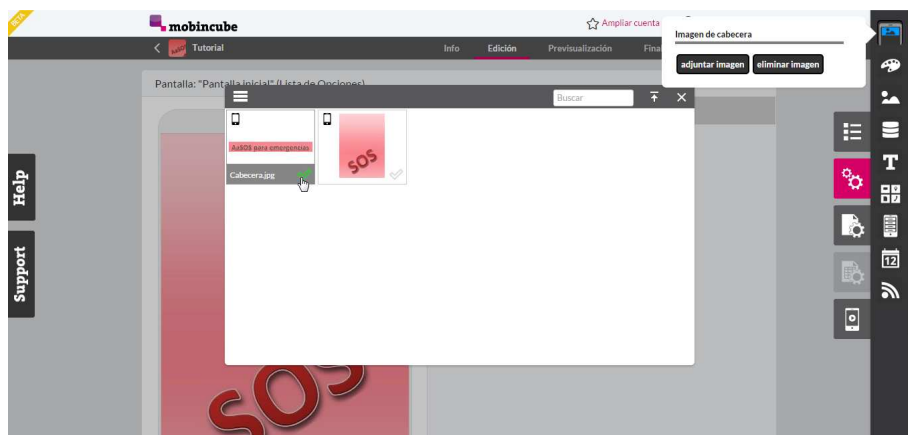


Ilustración 4.97: Tutorial Mobincube – Selección de imagen de cabecera, paso 6

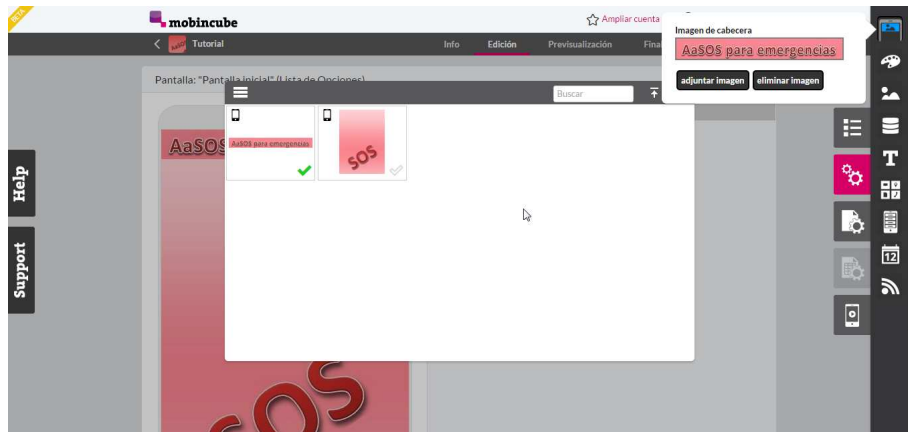


Ilustración 4.98: Tutorial Mobincube – Selección de imagen de cabecera, paso 7

- Paso 8: Añadir opción de menú “Emergencias 112”.



Ilustración 4.99: Tutorial Mobincube – Añadir opción de menú



Ilustración 4.100: Tutorial Mobincube – Añadir opción de menú “Emergencias 112”

- Paso 9: Asignar acción al botón

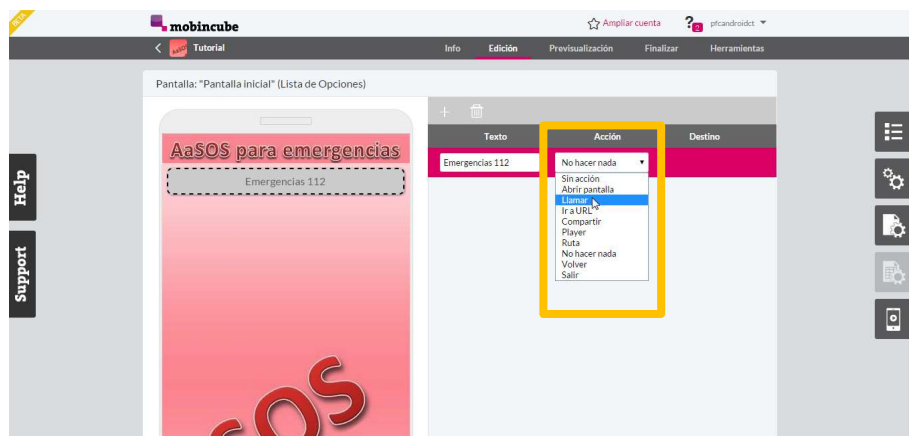


Ilustración 4.101: Tutorial Mobincube – Asignar acción al botón “Emergencias 112”, paso 1

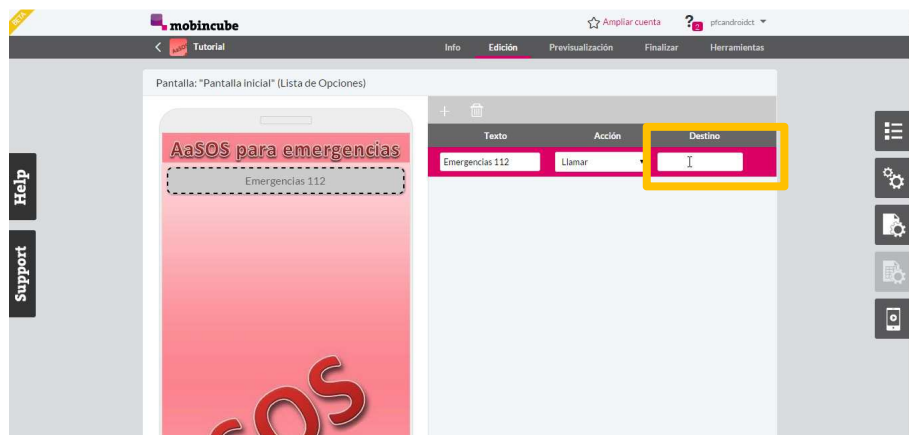


Ilustración 4.102: Tutorial Mobincube – Asignar acción al botón “Emergencias 112”, paso 2

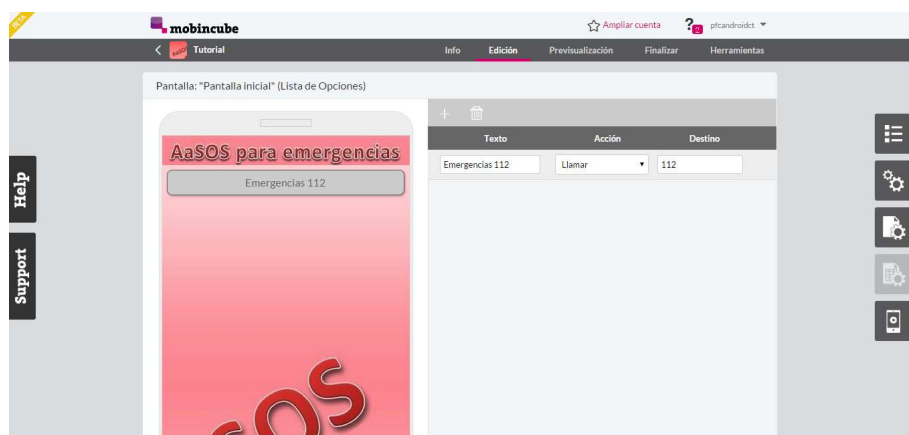


Ilustración 4.103: Tutorial Mobincube – Asignar acción al botón “Emergencias 112”, paso 3

- Paso 10: Ajustar propiedades de los botones con los valores deseados. Esta configuración se aplicará a todos los botones de la lista de opciones.



Ilustración 4.104: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 1

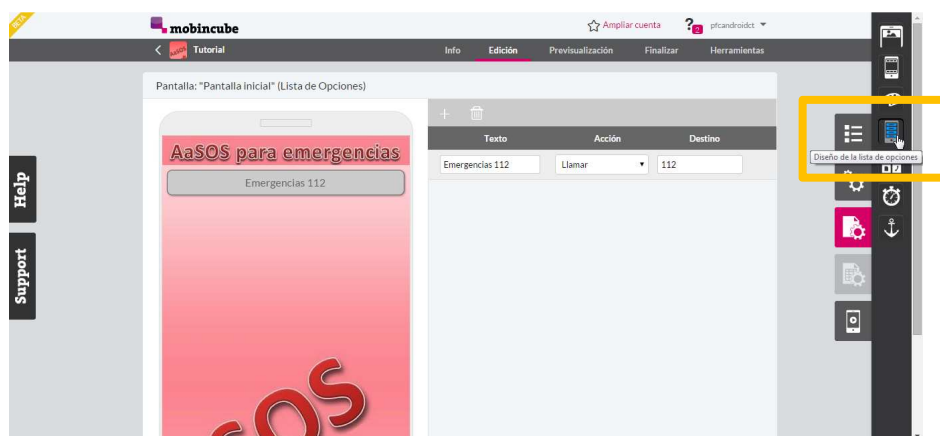


Ilustración 4.105: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 2

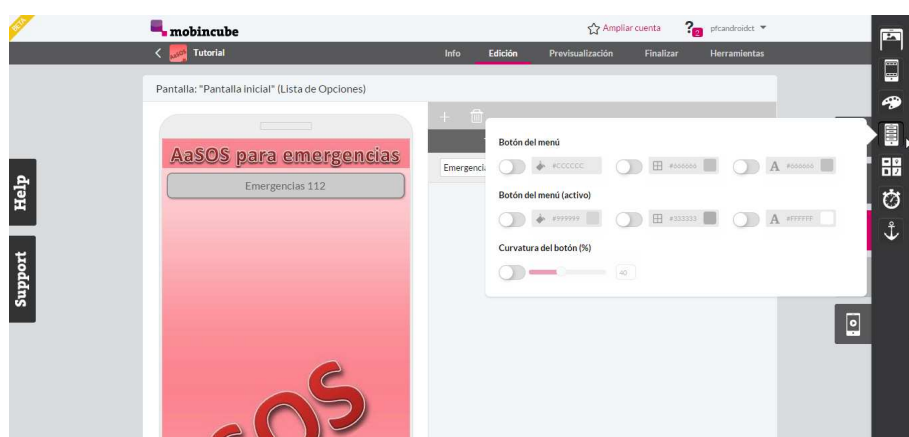


Ilustración 4.106: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 3

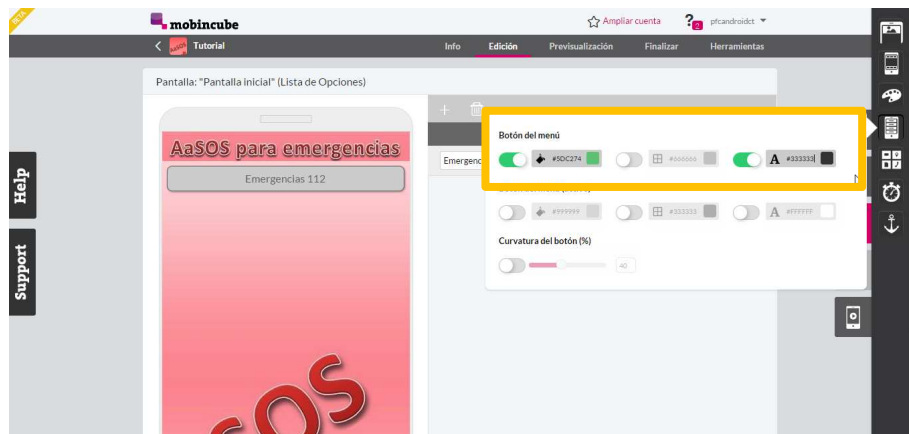


Ilustración 4.107: Tutorial Mobincube – Ajustar propiedades de la lista de opciones, paso 4

- Paso 11: Añadir botón “Guardia Civil”.



Ilustración 4.108: Tutorial Mobincube – Añadir opción de menú “Guardia Civil”, paso 1



Ilustración 4.109: Tutorial Mobincube – Añadir opción de menú “Guardia Civil”, paso 2

- Paso 10: Añadir resto de opciones de menú del mismo modo que los anteriores y configurar sus acciones asociadas.



Ilustración 4.110: Tutorial Mobincube – Añadir todas las opciones del menú, paso 1

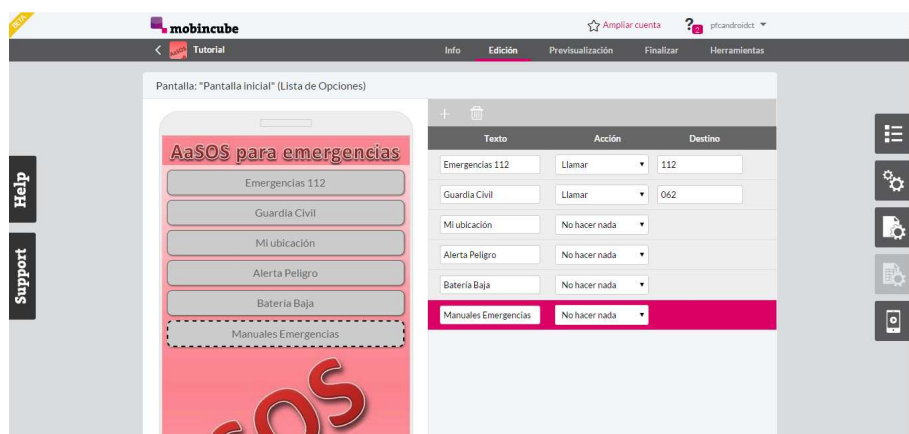


Ilustración 4.111: Tutorial Mobincube – Añadir todas las opciones del menú, paso 2

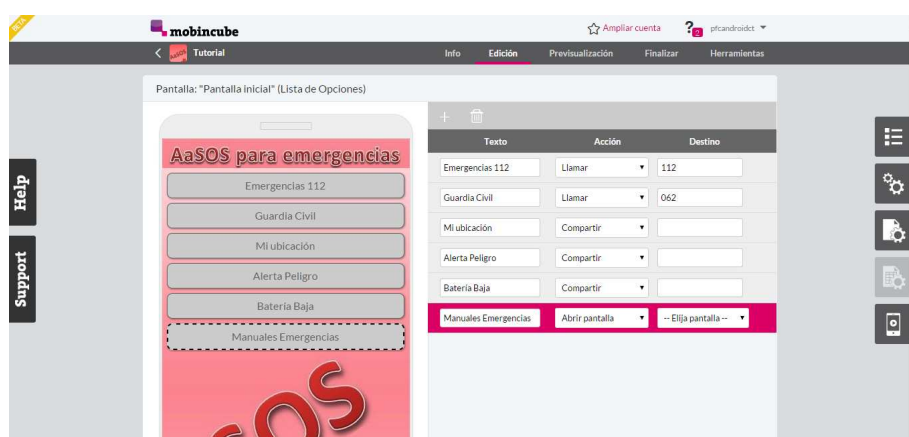


Ilustración 4.112: Tutorial Mobincube – Añadir todas las opciones del menú, paso 3

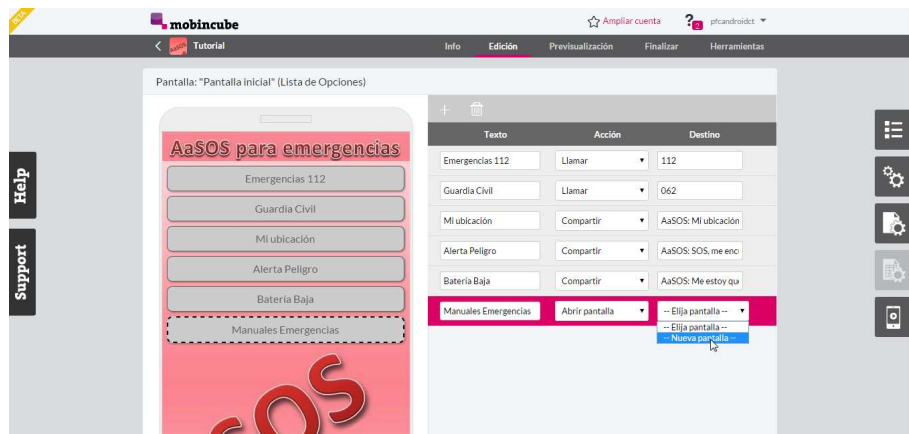


Ilustración 4.113: Tutorial Mobincube – Añadir todas las opciones del menú, paso 4

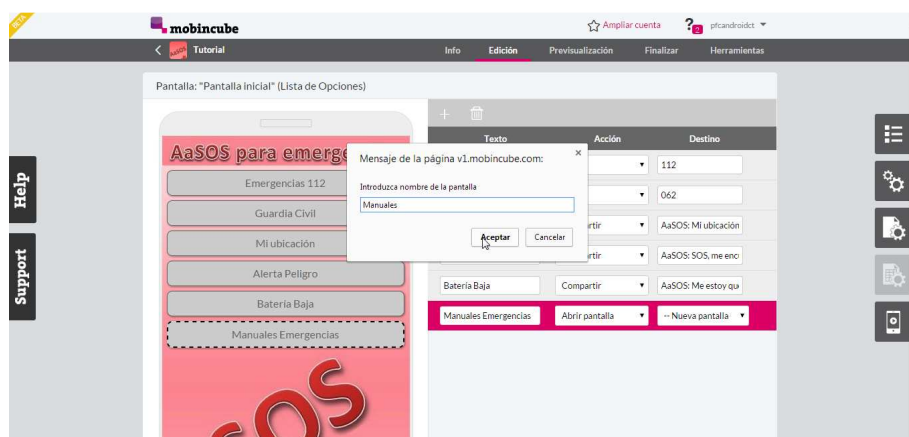


Ilustración 4.114: Tutorial Mobincube – Añadir todas las opciones del menú, paso 5

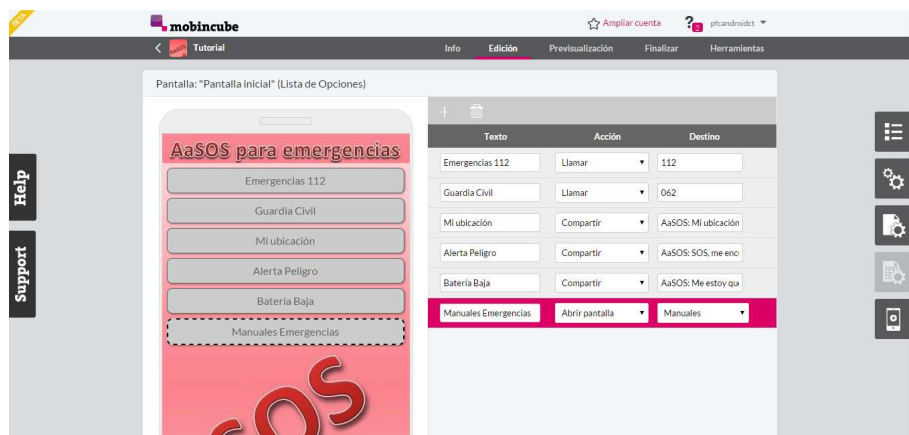


Ilustración 4.115: Tutorial Mobincube – Añadir todas las opciones del menú, paso 6

Conclusiones

Después de todo lo expuesto, son varias las conclusiones que se pueden extraer del estudio:

- Es necesario tener siempre presente las limitaciones técnicas con las que cuentan las herramientas generadoras de código Android. Estas restricciones vienen provocadas por el sencillo manejo de las mismas y por los prácticamente nulos conocimientos y medios técnicos requeridos al usuario.
Como producto de dichas limitaciones resulta el escaso alcance, en cuanto a funcionalidad y diseño se refiere, que poseen las aplicaciones realizadas.
- Resulta factor clave la elección de la herramienta a utilizar, tanto porque la misma tenga las funcionalidades suficientes o necesarias para realizar la aplicación que se desee, como por el nivel de conocimientos en programación Java orientada a objetos que el usuario posea.
- Estas herramientas ponen al alcance de prácticamente cualquier persona la realización de una aplicación móvil de forma totalmente gratuita. Ofrecen también la posibilidad de llegar más lejos, pudiendo crear, publicar y gestionar aplicaciones de cierta envergadura; para esta posibilidad es probable tener que invertir algo de dinero, bien para la adquisición de paquetes que amplían y mejoran la herramienta y/o bien para la publicación de la App realizada.

Como valoración personal puedo decir que es difícil, por no decir imposible, que estas herramientas lleguen algún día a alcanzar el nivel y la calidad de una aplicación creada con código fuente en un entorno de desarrollo por una persona cualificada para ello.

Me gustaría resaltar en cambio, la oportunidad que estas herramientas brindan a la gente tanto de poder inmiscuirse algo en el mundo de las Apps, como de poder tener una aplicación de forma gratuita (factor importante en los tiempos de crisis en los que nos encontramos) para usos tan útiles como fomentar su negocio, obtener ingresos si se consigue vender o simplemente facilitar en algún aspecto la vida de esa o de varias personas.

Anexos

ANEXO I: Diseño de AaSOS con MIT App Inventor 2

➤ Vista de Diseño

- Pantalla de descarga de Pdf's

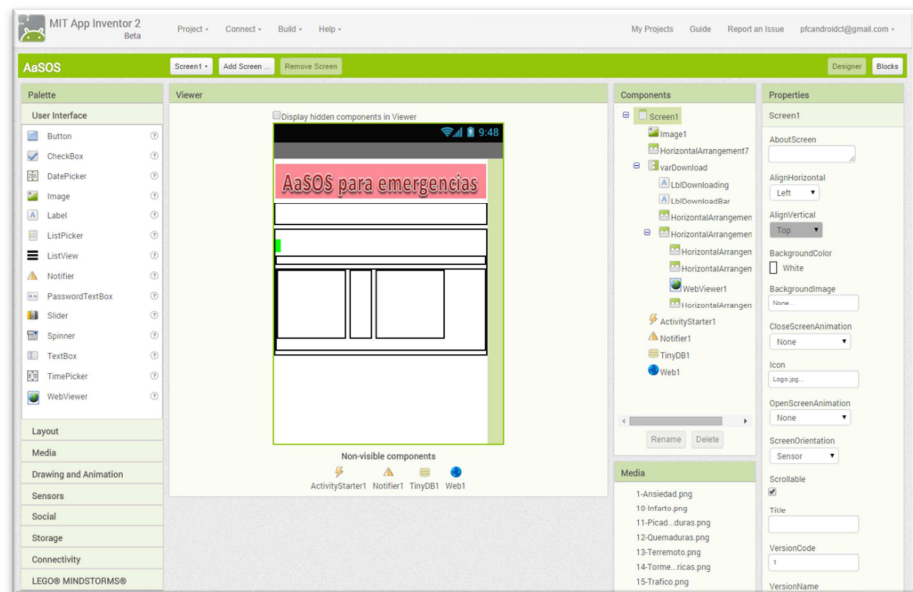


Ilustración Anexo I - 1: MIT App Inventor 2 - Vista de diseño de pantalla inicial

- Pantalla de menú

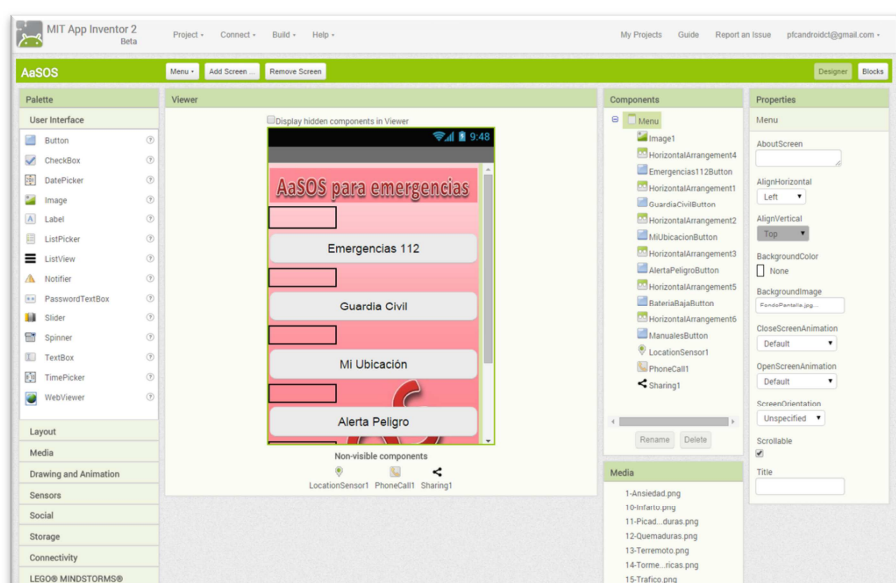


Ilustración Anexo I - 2: MIT App Inventor 2 - Vista de diseño de pantalla de menú

- Pantalla de listado de manuales

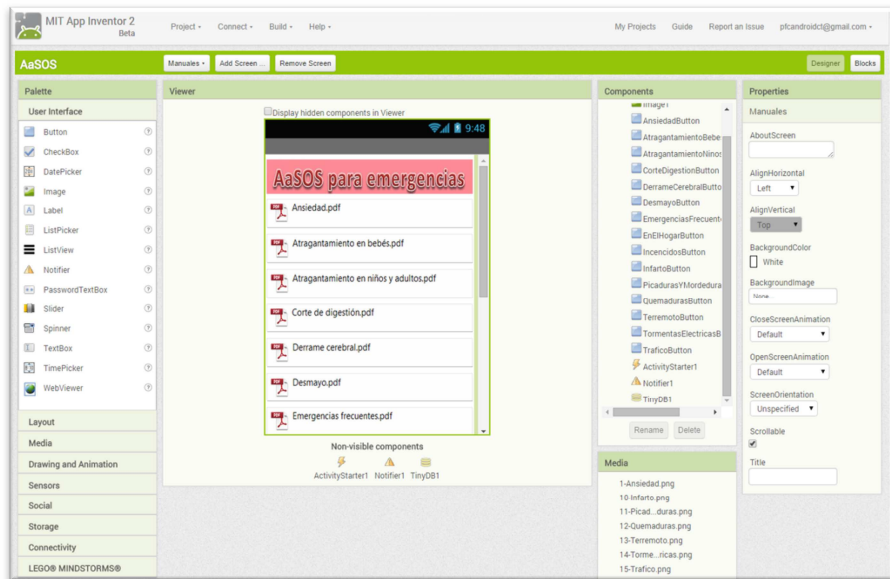
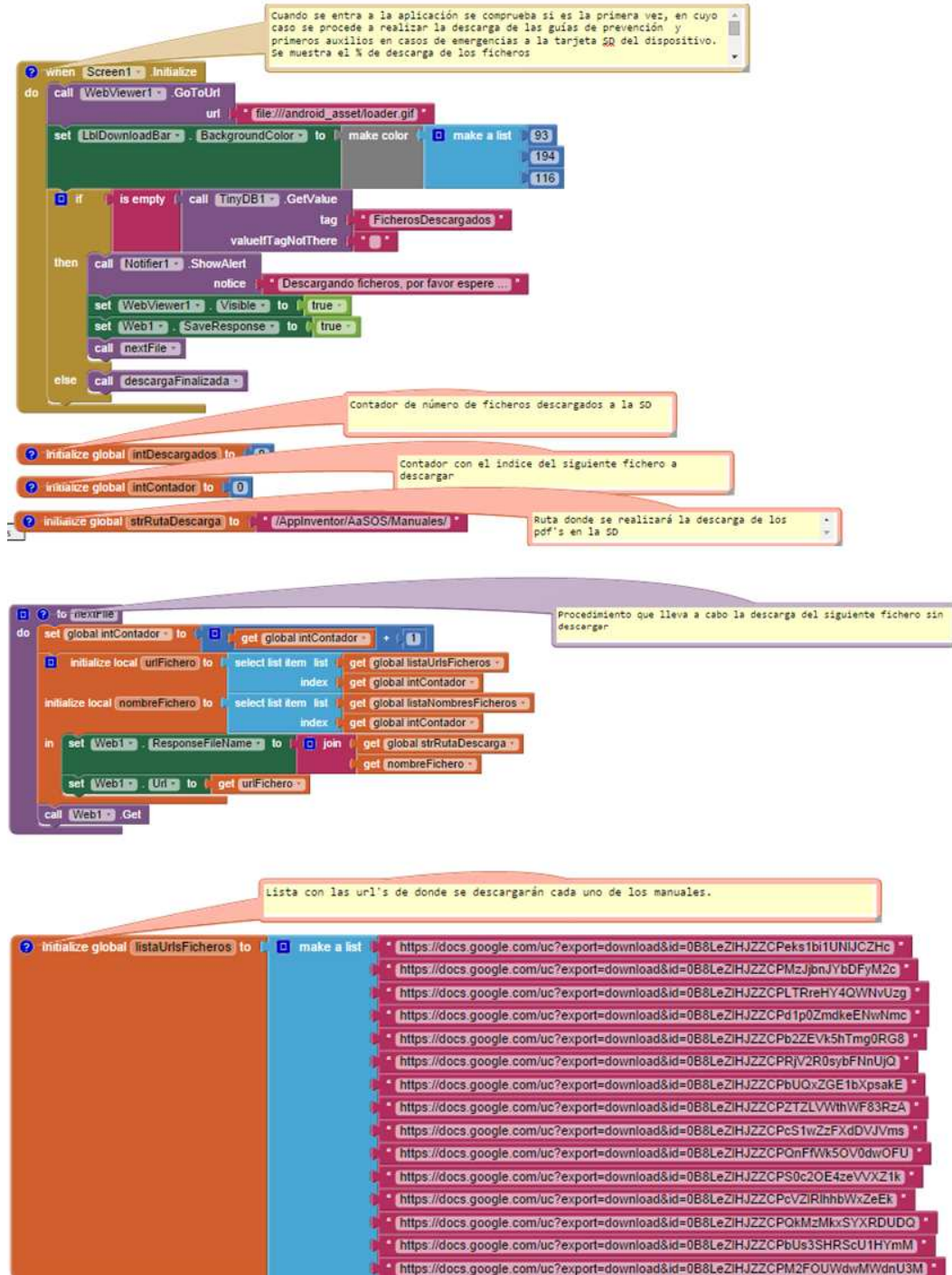


Ilustración Anexo I - 3: MIT App Inventor 2 - Vista de diseño de pantalla listado manuales

➤ Vista de Bloques

- Pantalla de descarga de Pdf's



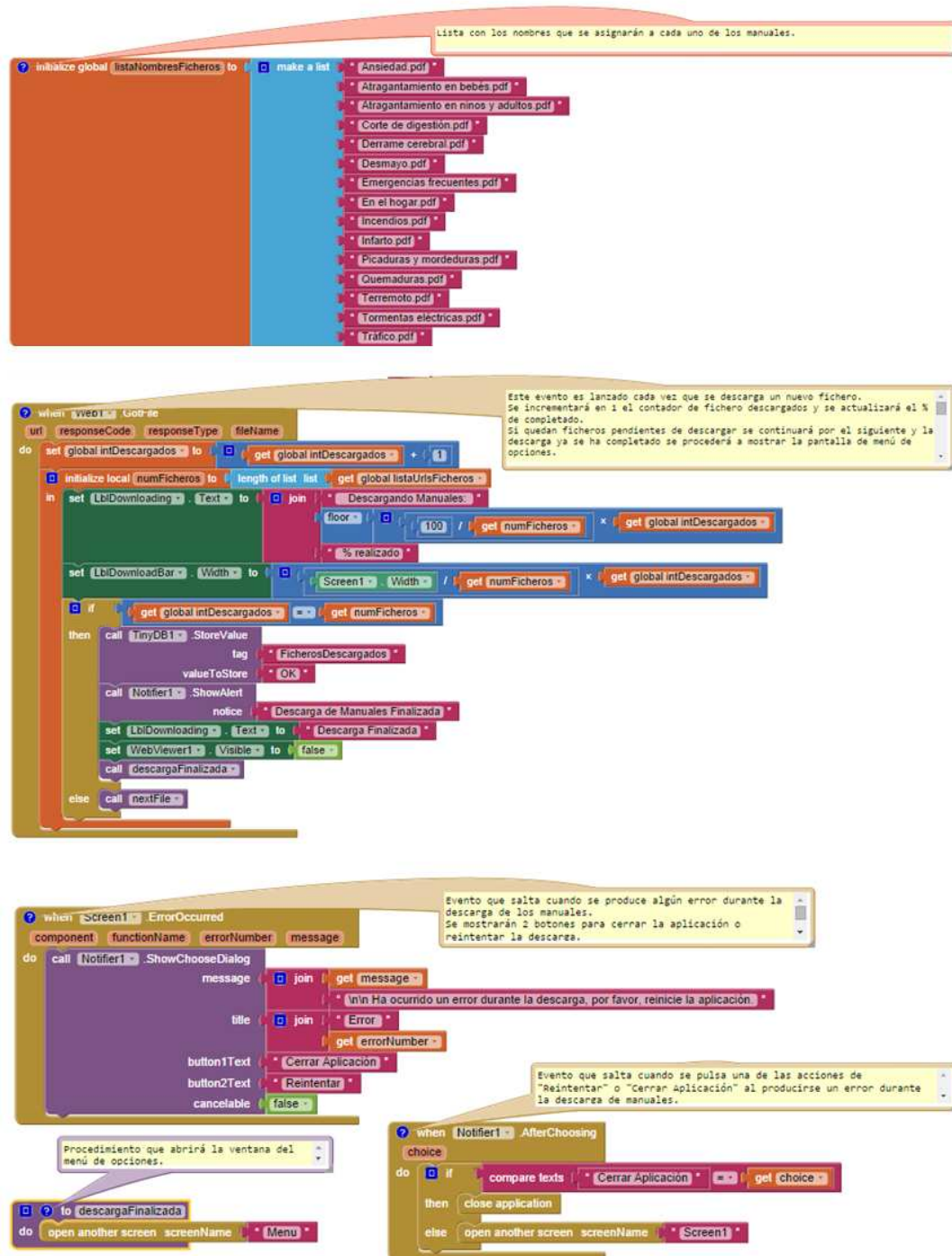
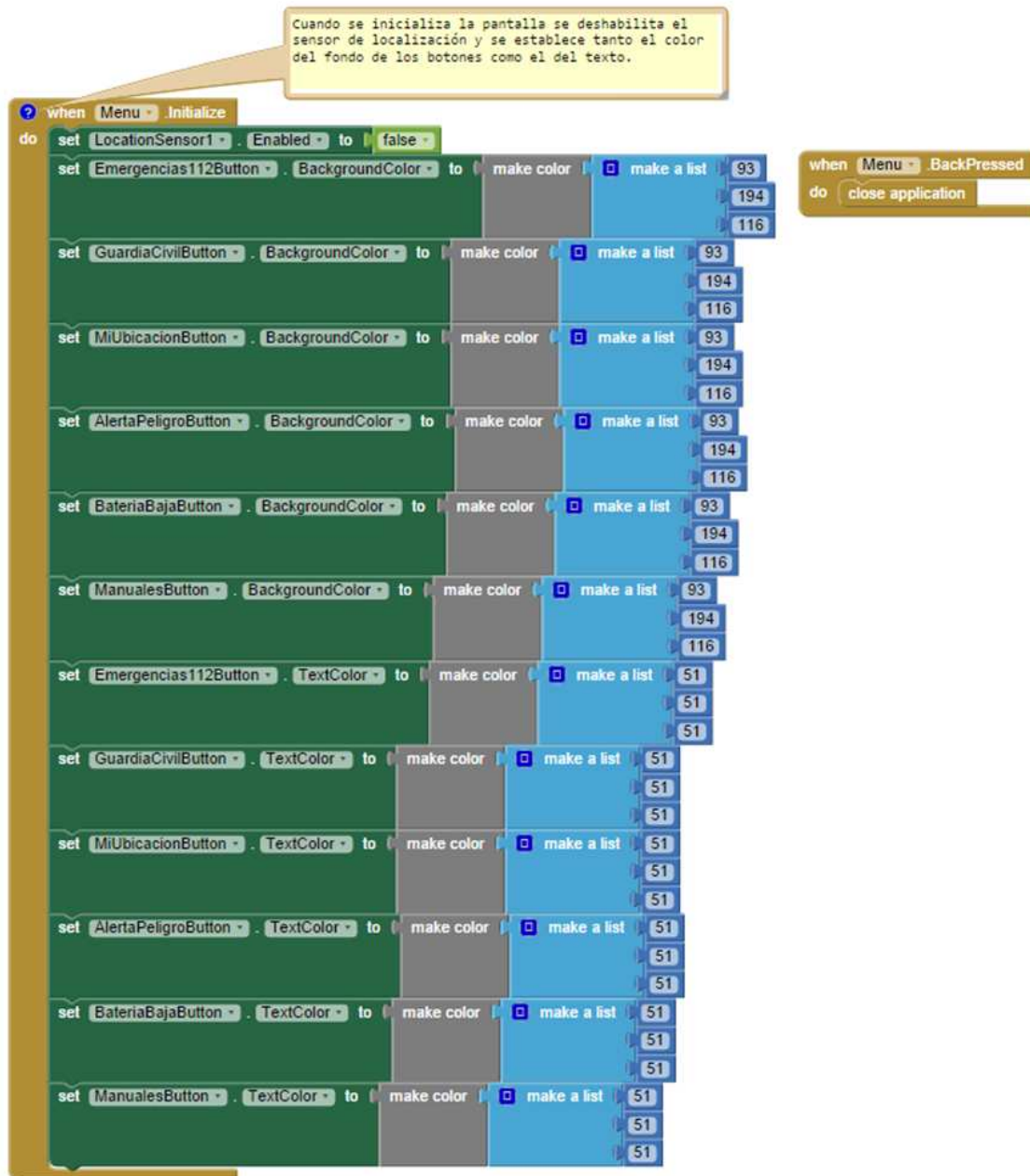


Ilustración Anexo I - 4: MIT App Inventor 2 - Vista de bloques de pantalla inicial

- Pantalla de menú



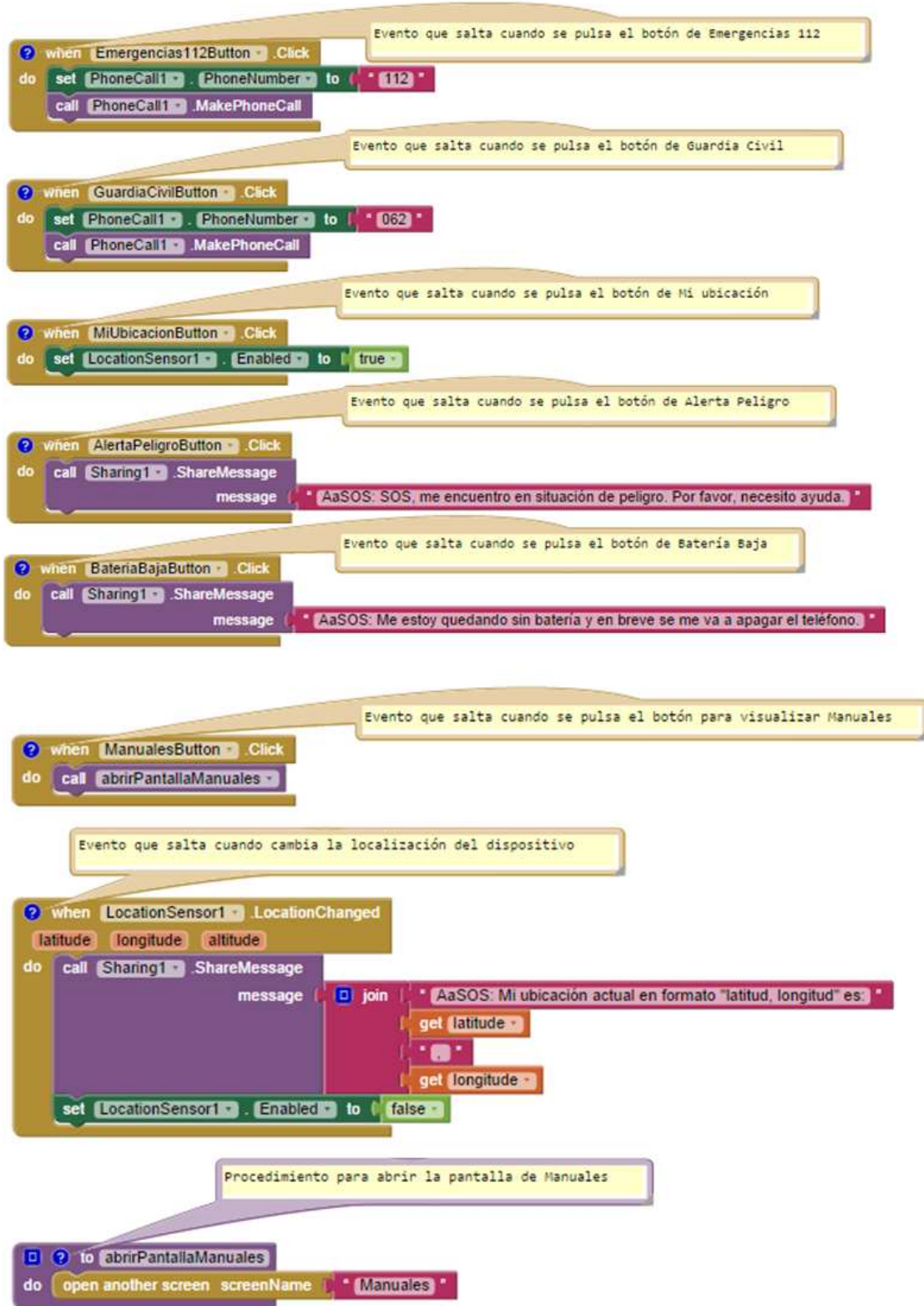


Ilustración Anexo I - 5: MIT App Inventor 2 - Vista de bloques de pantalla de menú

- Pantalla de listado de manuales



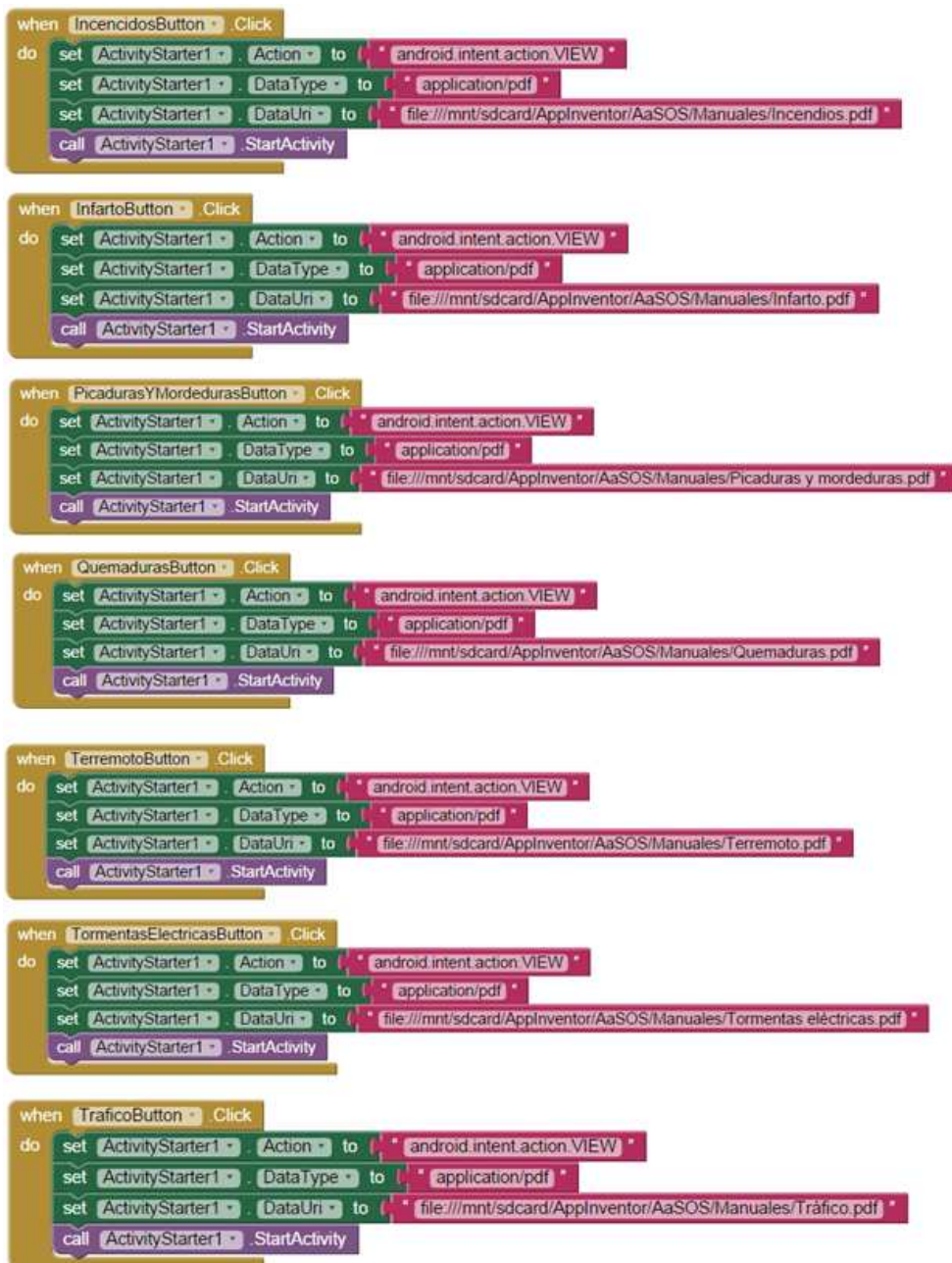


Ilustración Anexo I - 6: MIT App Inventor 2 - Vista de bloques de pantalla listado manuales

ANEXO II: Diseño de AaSOS con Mobincube

- Pantalla de menú

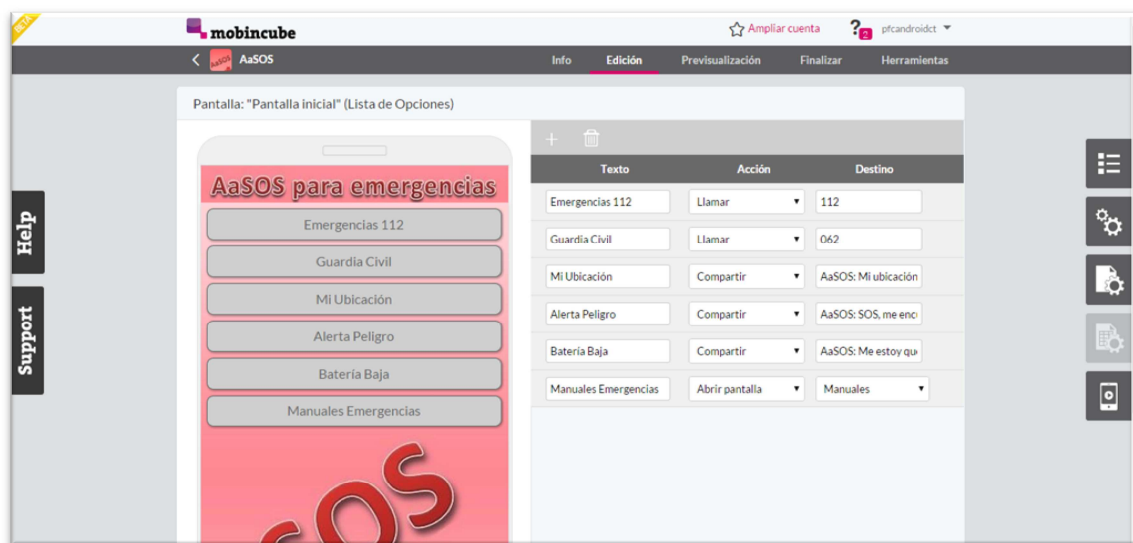


Ilustración Anexo II - 1: Mobincube – Diseño de pantalla de menú

- Pantalla de listado de manuales

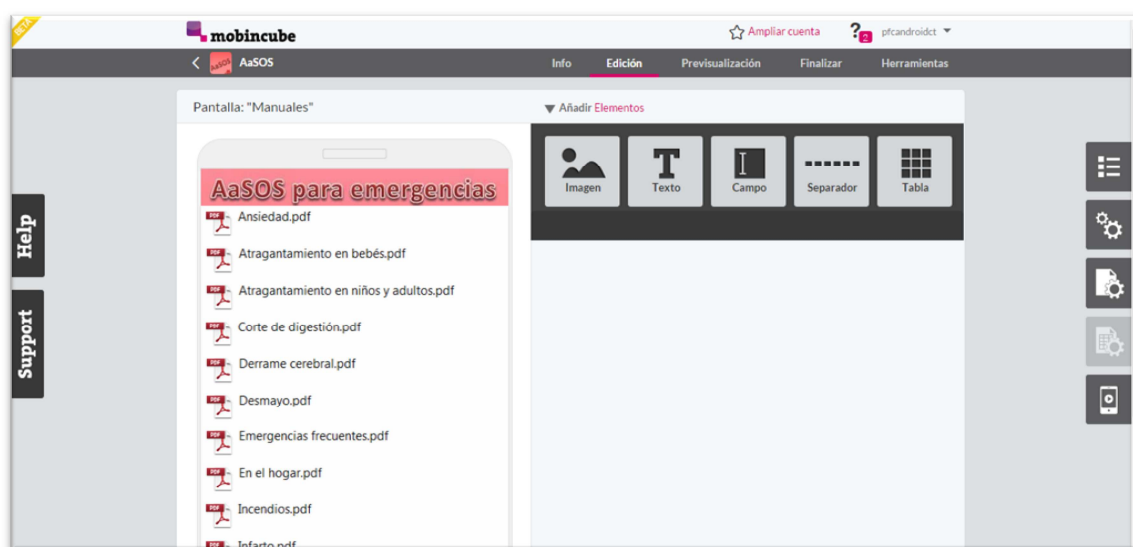


Ilustración Anexo II - 2: Mobincube - Diseño de pantalla listado manuales

- Acción al pulsar sobre un manual.

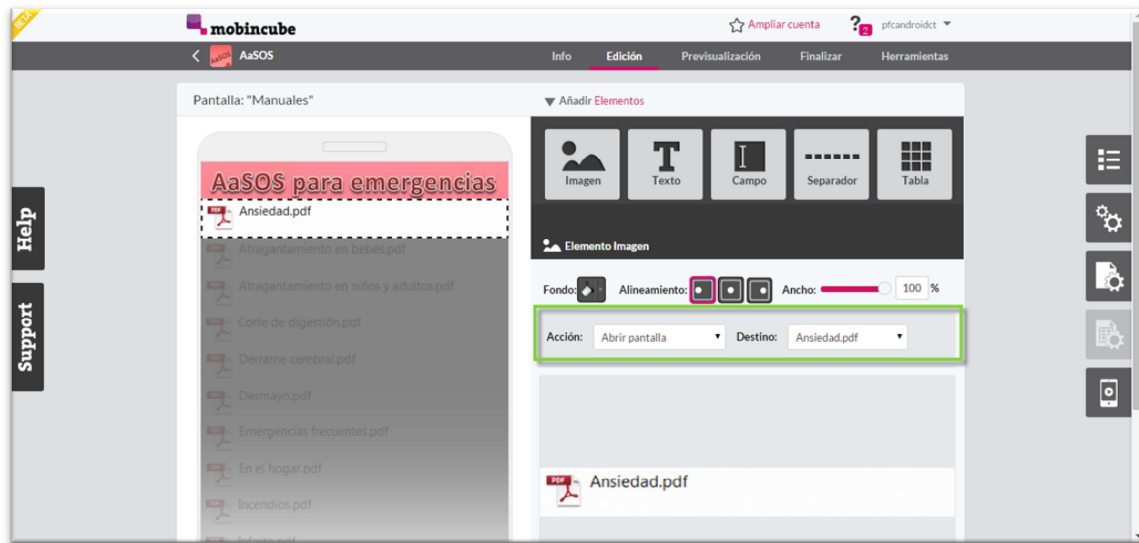


Ilustración Anexo II - 3: Mobincube - Diseño de acción abrir manual

- Pantalla de visualización de manual.

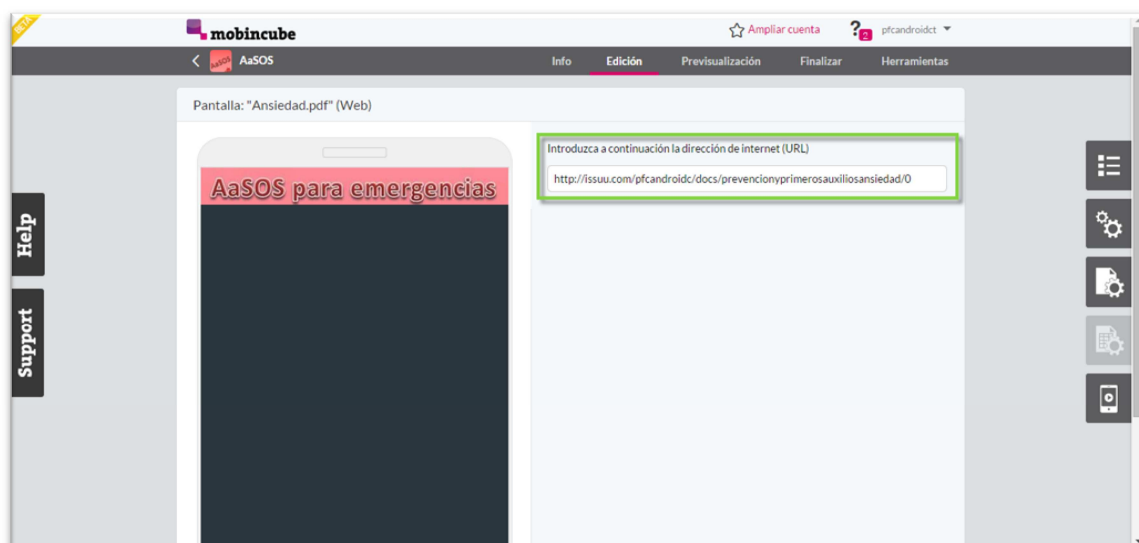


Ilustración Anexo II - 4: Mobincube - Diseño de pantalla para visualización de manual

Referencias

- MIT App Inventor → <http://appinventor.mit.edu/explore/>
- Mobincube → <http://www.mobincube.com/es/>
- Buzztouch → <https://www.buzztouch.com/>
- Adiante apps → <http://www.adianteapps.com/inicio>
- Andromo → <http://andromo.com/>
- Appmakr → <http://www.appmakr.com/>
- Apps Builder → <http://apps.apps-builder.com/dashboard/manager>
- Apps Geyser → <http://www.appsgeyser.com/es/>
- Appyet → <http://www.appyet.com/>
- Creapp → <http://creapp.es/>
- Game Salad → <http://gamesalad.com/>
- Ibuildapp → <http://ibuildapp.com/>
- Nimbo solutions → <http://www.nimbosolutions.com/>
- Octopus → <https://octopusapps.com/es/>
- Rheti → <http://www.rheti.com/>
- Shoutem → <https://www.shoutem.com/>
- Wiiappii → <http://www.wiiappii.com/>